Masters Program in **Geospatial Technologies**

WIFI INDOOR POSITIONING FOR MOBILE DEVICES, AN APPLICATION FOR THE UJI SMART CAMPUS

JOAN PERE AVARIENTO VICENT

Dissertation submitted in partial fulfilment of the requirements for the Degree of *Master of Science in Geospatial Technologies*

UNIVERSIDADE NOVA ISEGI
UNIVERSITAT JAUME I
Westfälische Wilhelms-Universität Münster

WIFI INDOOR POSITIONING FOR MOBILE DEVICES, AN APPLICATION FOR THE UJI SMART CAMPUS

Dissertation supervised by

PhD Francisco Ramos Romero

PhD Michael Gould Carlson

MSc André Figueiredo Barriguinha

March 2013

ACKNOWLEDGMENTS

This research project would not have been possible without the support of many people.

Prof. Dr. Francisco Ramos who was abundantly helpful and offered invaluable assistance, support and guidance.

Deepest gratitude are also due to the members of the supervisory committee, Dr. Michael Gould and MSc. André Figueiredo.

Special thanks also to the Assoc. Prof. Andrés Muñoz who was offered some help during the project development of the work.

Special thanks also to all my graduate friends by all experiences together throughout the master specially to Alberto Olivares for his support in some difficult moments and for his friendship.

Not forgetting Paula Oliete who always been there.

Last but not the least, I would like to thank my family: my parents Pedro Avariento and Rosa Mª Vicent, and my sister Marina Avariento for supporting me throughout my life.

"I have not failed. I've just found 10,000 ways that won't work."

Thomas Alva Edison

"If I had eight hours to chop down a tree, I'd spend six sharpening my ax."

Abraham Lincoln

WIFI INDOOR POSITIONING FOR MOBILE DEVICES, AN APPLICATION FOR THE UJI SMART CAMPUS

ABSTRACT

Smart Campus has emerged as a study platform of a Smart City. There are some similarities between the Campus and a City. Due to this, is possible use a Smart Campus as attesting bench and then apply these investigations to implement in a Smart City. One of this potential technologies is the Indoor Positioning System using the Wi-Fi network. The aim of this work is research and implement a mobile application to carry out the indoor positioning in the context of the UJI Smart Campus. The prototype developed allows to perform the first part of the Wi-Fi Indoor Positioning, the mapping phase. This application implements a system to display and all UJI cartography (campus basemap, UJI buildings and UJI buildings interiors). When whole system will be developed, it will allow implement the indoor positioning in a future applications for the Smart Campus.

KEYWORDS

Smart City

Smart Campus

GIS Applications

Geographical Information System

Wi-Fi Indoor Positioning System

Wireless positioning

Indoor Positioning

Indoor Localization

Smart Phones

Android Mobile Application

ArcGIS Runtime SDK for Android

ACRONYMS

A-GPS	Assisted GPS (see GPS)
AOA	Angle of Arrival
AP	Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identifier, same as MAC in value
Cell-ID	Cell Identification
dBm	Decibel miliwatt
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ICT	Information and Communication Technologies
IEEE	Institute of Electrical and Electronics Engineers
INIT	Institute of New Imaging Technologies
LAN	Local Area Network
LOS	Line of Sight
MAC	Medium/Media Access Control, same as SSID in value
REST	Representational State Transfer
RFID	Radio Frequency Identifier
RSS(I)	Received Signal Strength (Indication)
SDI	Spatial Data Infrastructure
SDK	Software Developer Kit
SQL	Structured Query Language
SS	Sum of Squares
SSID	Service Set Identifier
TDOA	Differential Time of Arrival
TOA	Time of Arrival
UI	User Interface
UJI	Universitat Jaume I
UMTS	Universal Mobile Telecommunications System
UNIX	Uniplexed Information and Computing System
URL	Uniform Resource Locator
US	United States
UWB	Ultra Wide Band
viscaUJI	Virtual Smart Campus for the Universitat Jaume I
WAP	Wi-Fi Access Point
Wi-Fi	Wireless Fidelity
WIPS	WiFi Indoor Positioning System
WLAN	Wireless LAN (see LAN)
XML	eXtendible Markup Language

TABLE OF CONTENTS

PART III: CONCLUSION

PART IV: FUTURE DIRECTIONS

PART V: BIBLIOGRAPHY

PART VI: ANEXES

INDEX OF TABLES

INDEX OF FIGURES

PART 0 - INTRODUCTION

CONTENTS

SUMMARY

In this part, it is going to do a introduction of the project by outlining and framing the project. It starts talking about the campus as a platform concept that it will briefly explain the concept of Smart Campus and then it will explains the importance and scope of the location based services applied as a Smart Campus service.

The second point of this part explains the goal and objectives of this work also the scope and limitations found.

1-. INTRODUCTION

A campus is a platform where there are people spending same hours at a day many days at week. These people have a different role in the campus (students, professors, campus staff and campus managers/administrators). This platform has different expenses (energy, maintenance, information, etc. expenses) and offers same services. For these reasons the campus platform and the city platform have several similarities.

Optimize the resources and improve the services is a key factor for the campus managers. Regarding the improvement of the services in a campus, about 80% of the data that the campus administration works with daily are georeferenced. In the same way, most of the campus information that the students use are georeferenced too.

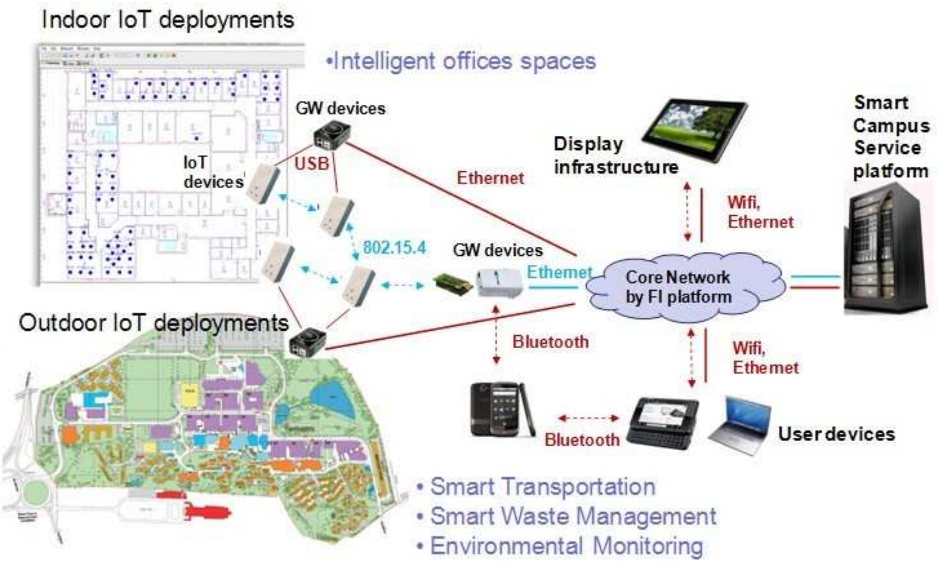


Figure 1: Outline the architecture of a smart campus (Oredope, Gluhak, & Evans)

The concept of Smart Campus appears in order to use the new technologies and apply it to improve the services and reduce the expenses. Smart Campus can improve all services and it's a powerful tool to manage all information in a campus: wastes, transports, access, classrooms and labs, energy expenditure, maintenance incidents…

Besides this is possible to use the smart campus as attesting bench and then apply these investigations to implement smart cities with tested applications and with a background.

Figure 2: Smart City technologies (Pareja, 2012)

The Smart UJI project is called viscaUJI project. This project is actually working and growing adding different kind of services and applications. Nowadays the project has a geodatabase (a database with geographic data and alphanumeric data) and different tools developed, most of them developed with web technologies. However the geodatabase needs add more data, such as buildings data, building interior data and other information that will be useful in the future.

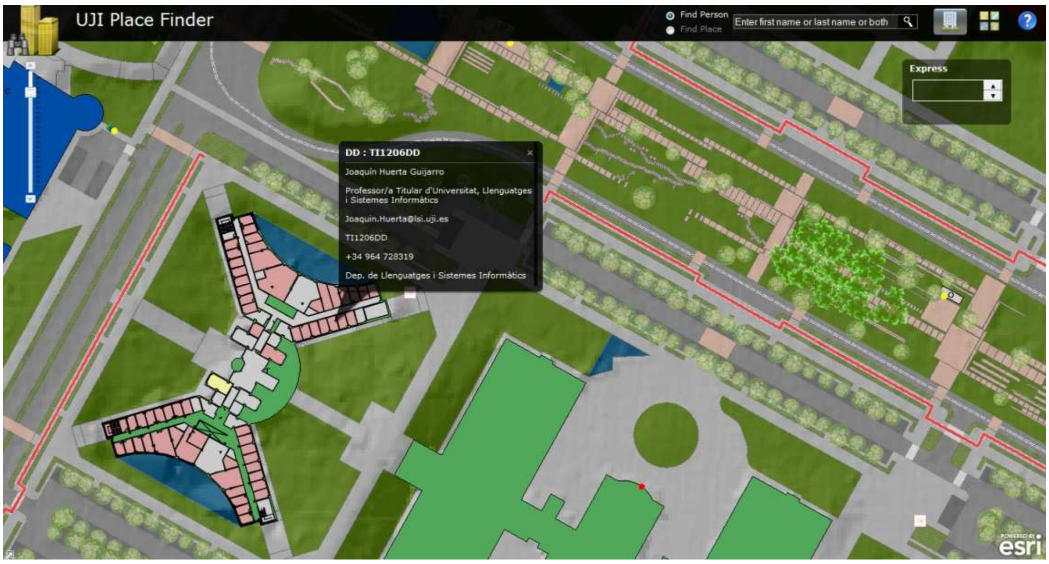


Figure 3: Place-Finder web-app of the viscaUJI project (Geotec)

An important part of any Smart Campus or any Smart City is the realizations of tools or application that use the new technologies. Nowadays the Smart Phones has been becoming a part and parcel of lives of most peoples. These are the reasons why Smart

Campus needs mobile application to provide to the Smart Campus user's tools and new services.

Technological advances within the past decade have caused a surge in the proliferation of personal locating technologies. Early consumer grade locating systems manifested as Global Position System (GPS) receivers fit for mounting on automobiles, aircraft, and watercraft. As computing and communication technologies have advanced, companies including Garmin Ltd., TomTom International, and Magellan Navigation Inc. have offered systems with increased usability and functionality. Current systems on the dashboard mounted, handheld, and wristwatch scales provide users the ability to determine their current location and find their way to their destination. Today’s advanced systems use measurements of signals from GPS, cellular communication towers, and wireless internet (Wi-Fi) access points to locate the user. Internet enabled mobile devices are becoming ubiquitous in the personal and business marketplaces. Integration of locating technologies into the Smart Phones has made the use of handheld devices that are dedicated to positioning obsolete. The availability of powerful communication and computing systems on the handheld scale has created many opportunities for readdressing problems that have historically been solved in other ways.

The signals used by outdoor locating technologies are often inadequate in this setting. Systems that rely on the use of cellular communication signals or identification of nearby Wi-Fi access points do not provide sufficient accuracy to discriminate between the individual rooms of a building. GPS based systems can achieve sufficient accuracy, but are unreliable indoors due to signal interference caused by walls, floors, furniture, and other objects.

Due to these limitations, and the need of this kind of applications by viscaUJI Smart Campus is necessary the study, research and implement an application able to positioning a user inside the buildings in the UJI campus. This application will be a basic and fundamental tool of the other applications such as navigation, place-finder, report maintenance breakdowns…

2-. GOAL/OBJECTIVES AND SCOPE/LIMITATIONS

The main objective of this work is research whether is possible to use the Wi-Fi technologies as a positioning technique for Smart Phones. This positioning technique will be developed to provide the position inside the UJI campus buildings.

The secondary objective is the implementation of a mobile application that will be a tool to perform the Indoor Positioning. This application will be a basic tool or a basic module to develop other applications that will need incorporate the indoor positioning such as navigation application, place finder, notice maintenance problems…

The framework of the application is the UJI Smart Campus project: viscaUJI. For this reason, the application must be compatible with the others mobile application of the whole project. In fact, all of these applications will be integrated in only one for the viscaUJI project, but this will be in the future.

For all of the above, the application must use the viscaUJI data (use the geodatabase of the project), and store the data in a compatible format probably because this data will be integrated to the main geodatabase in the future.

The academic objective of this work is performing the Master's Thesis of the Master in Geospatial Technologies. Consequently it is a requisite perform the work in a few months.

PART I – THEORETICAL FRAMEWORK

CONTENTS

SUMMARY

This part explains the theoretical framework of the project. First a smart campus is explained in detail also the mobiles platform. By other hand different sensor techniques and positioning techniques that a mobile device can use are explained at the point 5.

3-. SMART CAMPUS

3.1. Smart Cities and Smart Campus

The XXI century is called to be the century of the cities because the urbanization advanced with giant steps (Shapiro, 2005). Also, optimize the resources will be an important thing in the cities. Smart cities purpose rethink about the cities from the sustainability point of view: adjust all of resources demand, reduce the waste, the wastage and losses (Hernández Muñoz, 2011) (Caragliu, Del Bo, & Nijkamp, 2009). A city is a platform where the people live and work, where the companies develop their business and in this context there are many services managed by the city hall.

For these reasons the public administrator, in this case the local administrator, has to think about the management of the evolution of cities models. From the viewpoint of the municipal managers, that provides services to the city, have a Smart City helps a better automatic and efficient management of the urban infrastructures and resources. This will provide some benefices: expenditure reduction and improvement of services.

The Smart Cities is a good solution to address these facts. The Smart City applies the informatics and communication technologies and tries to create a more interactive and efficient infrastructure and their components and services.

The Smart City becomes a digital platform that maximizes the economy, society, environment and welfare of the cities, and facilitates the shift to more sustainable behavior among all stakeholders: users, businesses and government. It also seeks to maximize public budgets, precisely thanks to the improvement of the processes involved in the city and its inhabitants (Fundación Telefónica, 2011). Partnerships and clear cooperation strategies among main stakeholders are needed in order to share research and innovation resources such as: experimental technology platforms, emerging ICT tools, methodologies and know-how, and user communities for experimentation on Future Internet technologies and e-service applications. Common, shared research and innovation resources as well as cooperation models providing access to such resources will constitute the future backbone of urban innovation environments for exploiting the opportunities provided by Future Internet technologies (Schaffers, Komninos, Pallot, Trousse, Nilsson, & Oliveira, 2011).

Smart City has to pay attention to the role of social and relational capital in urban development. People need to be able to use the technology in order to benefit from it and for the own city (Caragliu, Del Bo& Nijkamp).

From the viewpoint of municipalities provided in the city, have a Smart City help the automatic and efficient management of urban infrastructure, which has obvious advantages: first, the reduction of expenditure and secondly the improvement of one's self services (Fundación Telefónica, 2011).

There are a parallelism between a city and a university campus and it is possible use these characteristics and applies the smart city concept in a campus. The campus is smaller than a city but it is more controlled. The concept of Smart Campus means an improvement in the management of the resources and the facilities in the frame of the campus. Due to these characteristics it is possible to use the Smart Campus as a testing bench (Sanchis, y otros, 2012).

The *Smart Campus* offers several improvements for whole campus community. The campus managers have several tools to manage, analyze and take decisions about the services, the resources and the campus facilities. The campus has to pay attention to the role of social in the campus. The students need to be able to use the technology in order to benefit from it and for the own campus (Caragliu, Del Bo& Nijkamp).

3.2. GIS in Smart Campus

About 80% of the data that the campus administration and campus staff works daily are georeferenced (Coll, Martínez, & Irigoyen, 2005). The same experience in these administrations have proved that the gradual incorporation of a Geographic Information System (GIS)in the municipal databases are the best way and the best tool to work with this data (Coll, Martínez, & Irigoyen, 2004) (Coll, Martínez, Ibarz, & Elgezaba, 2007) (Martínez, Coll, & Irigoyen, 2004) (Peñarrubia & Rubio, 2007).

However, there are more problems in the administration related with the spatial data: the campus staff doesn't know well the spatial data; the spatial and numerical data can't use together because is stored in incompatibles formats; there aren't communication between data of the different departments; there are redundant data; the numerical data is not related with his spatial data (Coll, Martínez, Ibarz, & Elgezaba, 2007).

The GIS provides solutions to visualize and integrate global and local geographical information. The Spatial Data Infrastructure (SDI), in the distributed version, provides the tools to access to the needed information in an interoperable way. These tools allow users to do analysis on the different systems on the smart campus (Sanchis, y otros, 2012). This local SDI in the Universitat Jaume I (UJI) is *viscaUJI*. The data is centralized and single.

Figure 4: Logo of Smart UJI

The *viscaUJI* project improves the resource management on the UJI campus. This project is based in a SDI local model and integrates all available information in a GIS. The viscaUJI GIS is the most powerful tool to manage the campus data and it is a tool to do the analysis of the different systems in a campus. This GIS is the best base to develop any kind of application or service for staff and students.

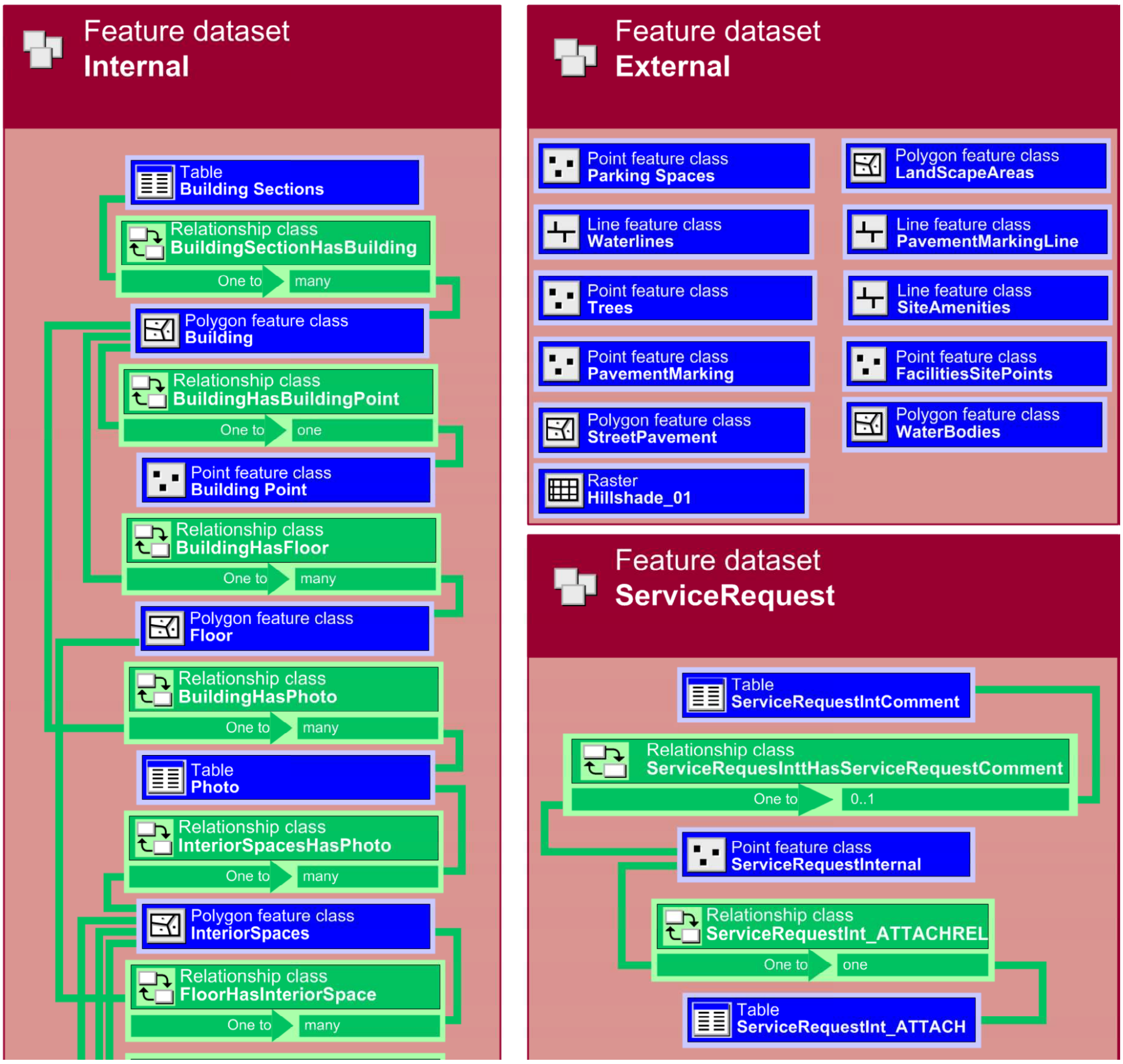


Figure 5: A version of UJI data model (Sanchis, et al., 2012)

ViscaUJI integrates all information (geographic and alphanumeric) following the ESRI Local Government Information Model (LGIM) (Sanchis, et al., 2012). The UJI smart campus enables a set of services to provide data to the applications that could use it. These services are offered through internet and there are:

- Web Map Service (WMS)
- Gazetteer Service
- Catalog Web Service (CWS)

4-. MOBILE PLATFORM

4.1. Introduction

A mobile platform is a portable device, typically a mobile phone or a smart phone, on which a user can perform different functionalities such as cellular networking and Internet access. In recent years, mobile phones have evolved greatly since the first devices, large and heavy, designed only for phoning anywhere. There are more and more new functions that are being added to the current generation of mobile device to increase their functionality and productivity. There are currently four popular basic platforms that are implemented directly or developed by other higher platforms: Symbian, UNIX, Apple and Windows. Among these platforms, only UNIX is an open source platform. Symbian and Windows are licensed by Nokia Inc. and Microsoft Inc., respectively. Apple develops and distributes the mobile operating system iOS. Next figure shows the mobile platforms that are actually in the market:

Figure 6: Mobile platforms (Poder PDA)

For each platform, Software Developer Kits (SDKs) are provided by the platform producer to help third party developers create applications for use on their mobile platform. To protect the closed source of the structure of the operating system, the developer is able to access different functionalities of the operating system through Application Programming Interfaces (APIs).

4.2. Android

The main operating system to target the application of this work are Android and iOS because there are the two main mobile operating systems. Next table shows a comparison between Android and iOS.

ANDORID	iOS
68% worldwide market of SmartPhones	17% worldwide market of SmartPhones
Is incorporated in multiple SmartPhones	iOS is only available by Apple
Allow custom and access whole system	iOS does not allow that

Table 1: Comparative between Android and iOS (Luna, 2012)

Android is the operating system selected to develop and implement this work. The main reasons for choosing Android are:

Android…

- is an easy programming language if one has previous knowledge of Java or C #.
- is open source.
- has an efficient management of the available resources.
- has a large community of developers and users.

Thus was born Android. Android is an operating system and a software platform based on Linux for Smartphone and tablets. Android lets you program in a working environment of Java applications. Moreover the most important difference between the others operating systems is that anyone who can program can create new applications, widgets, or even modifies the operating system itself because is open source.

Figure 7: Android logo (Wikipedia)

4.3. ArcGIS SDK for Android

ArcGIS Runtime SDKs for Smart Phones enable to develop users to build applications that utilize the powerful mapping, geocoding, geoprocessing and custom capabilities provided by ArcGIS for server. Using this SDK, the user has the ability to embed ArcGIS maps and tasks into the application.

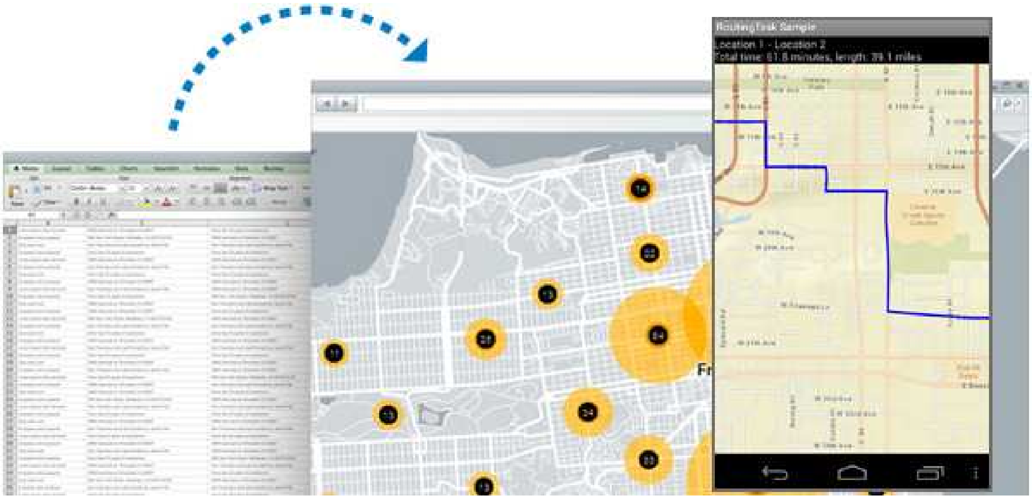


Figure 8: ArcGIS Runtime SDK for Android (ESRI)

The most powerful capabilities that ArcGIS SDK provides to the applications are the follow:

- Use and display services from ArcGIS Online and/or ArcGIS for Server.
- Execute sophisticated geoprocessing tasks and displays results.
- Create applications that collect data and update data.
- Use a rich set of tasks which leverage the power of ArcGIS to analyze the maps in the application and provide information to the application's users.
- Work with the device GPS mobile.
- Perform advanced geometric operations locally.

5-. SENSORS AND POSITIONING TECHNIQUES

5.1. Available sensors technologies

There are many Location Based Services running on the Smart Phones, but there is a needed of a precise indoor localization system, since most of these applications rely on GPS (Martin, Vinyals, Friedland, & Bajcsy, 2010). There are many potential technologies integrated in a mobile platform that it is possible use to carry out the positioning. These technologies are GPS, Cell Tower, Wi-Fi, Bluetooth and Infrared.

The Bluetooth and Infrared technologies have been discarded because the high cost of implementation, the low range and the low range (Le, 2009).

A Cellular Communication Network is a system that allows mobile phones to communicate with each other. This system uses large cell towers to wirelessly connect mobile devices. The range of cellular communication networks depends on the density of large buildings, trees and other possible obstructions. Maximum range for a cell tower is 35 kilometers in an open rural area. This method is a basic technique using Cell-ID, also called Cell of Origin, to provide location services for cell phone users (Zeimpekis, Giaglis, & Lekakos, 2003). This method is based on the capability of the network to estimate the position of a cell phone by identifying the cell tower that the device is using at a specific time. The advantage of this technique is its ubiquitous distribution, easy implementation and the fact that all mobile cell phones support it. The accuracy of this technique is very low due to the fact that cell towers can support ranges of 35 kilometers or more. In urban environments cell towers are distributed more densely.

The GNSS (Global Navigation Satellite System) provide geo-spatial positioning with a global coverage. Currently there are several global navigation satellite systems dedicated to the civil positioning including the US NAVSTAR GPS (Global Positioning System), the Russian GLONASS, and the European Union's Galileo (Thurston, 2002). The advantage of satellite systems is that receivers can determine latitude, longitude and altitude to a high degree of accuracy. However, LOS (Line Of Sight) is required for the functioning of these systems. This leads to an inability to use these systems for an indoor environment where the LOS is blocked by walls and roofs.

Wireless Fidelity (Wi-Fi) is the common nickname for the IEEE 802.11 standard. Wireless connectivity is more prevalent than ever in our everyday lives. Each wireless router broadcasts a signal that is received by devices in the area. Wireless devices have the capability to measure the strength of this signal. This strength is converted to a number, known as *received signal strength indicator* (*RSSI*) (Le, 2009) (Halder & Kim, 2012). A user's device can detect the *RSSI* and *MAC* address of multiple routers at one time.

RSSI is a dimensionless metric that is used by systems to compare the strength of signals from multiple access points. There is no standard conversion between RSSI and the actual *received signal strength* (*RSS*). Important characteristics of *RSSI* to *RSS* conversions include: The maximum and minimum *RSSI* values (dimensionless integers), the maximum and minimum *RSS* values that can be represented (*dBm*), and the resolution of the conversion (value in *dBm* represented by one *RSSI* unit), (Le, 2009).

An advantage of Wi-Fi is that wireless networks are universal. They exist in population-dense areas and are continuously spreading outward. This causes Wi-Fi based systems to have a lower cost of implementation (Le, 2009).

5.2. Positioning technologies

Positioning techniques can be implemented in two ways: Self-positioning and remote positioning (Zeimpekis, Giaglis, & Lekakos, 2003). In the first approach, the mobile terminal uses signals, transmitted by the gateways/antennas to calculate its own position. In the second case (remote positioning) the mobile terminal can be located by measuring the signals travelling to and from a set of receivers. The receivers which can be installed at one or more locations measure a signal originating from the object to be positioned. These signal measurements are used to determine the length and/or direction of the individual radio paths, and then the mobile terminal position is computed from geometric relationships.

5.2.1. Self positioning techniques

GPS and Assisted GPS (A-GPS): GPS is the worldwide satellite-based radio navigation system. The system's satellites transmit navigation messages, which a GPS receiver uses to determine its position. GPS receiver processes the signals to compute position in 3D. As far as the A-GPS method is concerned, the mobile network or a third party service provider can assist the handset by directing it to look for the specific satellites and also by collecting data from the handset to perform location identification calculations that the handset itself may be unable to perform due to limited processing power. The A-GPS method can be extremely accurate, ranging from 1 to 10 meters (Zeimpekis, Giaglis, & Lekakos, 2003). Therefore GPS is largely unchallenged in the outdoor positioning domain where there is a clear view of the sky. There is still a need for other positioning technologies to remedy the serious shortcomings of GPS technology (Wang, Jia, Lee, & Li, 2003).

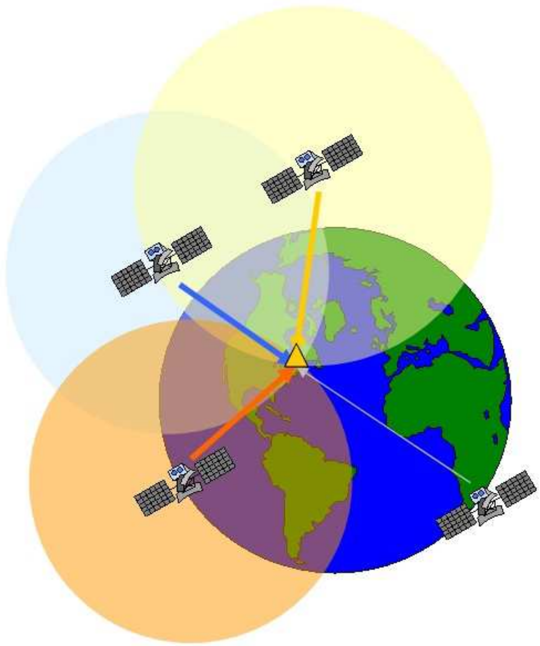


Figure 9: Scheme of GPS system

GPS is a semi-accurate global positioning and navigation system for outdoor applications (Lenihan, 2004). The accuracy of GPS devices is still in the range of 5-6 meters in open space. A GPS device cannot be used for an indoor environment because the LOS is blocked.

Methods have been developed to overcome the LOS requirement of GPS by setting up pseudolite systems that imitate GPS satellites by sending GPS-like correction signals to receiver within the building. A system has been developed by the Seoul National University GPS Lab, which achieves sub-centimeter accuracy for indoor GPS navigation system (Kee, Yun, Jun, Parkinson, Pullen, & Lagestein, 2001). This system has a convergence time of under 0.1 seconds, which helps to increase the responsiveness for a mobile user. This system uses pseudolites and a reference station to assist a GPS mobile vehicle in an indoor environment. The pseudolites have a fixed position and use an inverse carrier phase differential GPS to calculate the mobile user's position. The reference station is also fixed and transmits carrier phase correction to the mobile user. The system faces several challenges including serious multipath propagation errors and strict pseudolite synchronization requirements. The multipath propagation is addressed through the use of a pulse scheme. Using a center pseudolite solves the synchronization problem. The prototype has achieved 0.14 cm static error and 0.79 cm dynamic error. However, this system is very financially costly to implement, due to the requirement for a large number of pseudolites.

Assisted GPS (A-GPS) is primarily used in cellular phones (Lenihan, 2004). The A-GPS method uses assistance from a third party service provider, such as a cell phone network, to assist the mobile device by instructing it to search for particular satellite. Also, data from the device itself is used to perform positioning calculations that might not otherwise be possible due to limited computational power. A-GPS is useful when some satellite signals are weak or unavailable. The cell tower provides information that assists the GPS receiver. When using A-GPS, accuracy is typically around 10-20 meters but suffers similar indoor limitations to standalone GPS (Lenihan, 2004).

Indoor Global Positioning System (Indoor GPS): This system focuses on exploiting the advantages of GPS for developing a location-sensing system for indoor environments. Indoor GPS solutions can be applicable to wide space areas where no significant barriers exist. Indoor GPS takes into account the low power consumption and small size requirements of wireless access devices, such as mobile phones and handheld computers. The navigation signal is generated by a number of pseudolites (pseudo-satellites). The signal is designed to be similar to the GPS signal in order to allow pseudolite-compatible receivers to be built with minimal modifications to existing GPS receivers (Zeimpekis, Giaglis, & Lekakos, 2003).

5.2.2. Remote positioning techniques

Cell Identification (Cell-ID): this method is the basic technique to provide location services and applications. The mobile networks can identify the approximate position of a mobile handset by knowing which cell site the device is using at a given time. The main benefit is that all devices support this technology but the accuracy of the method is generally low (200 meters) depending on cell size (Zeimpekis, Giaglis, & Lekakos, 2003).

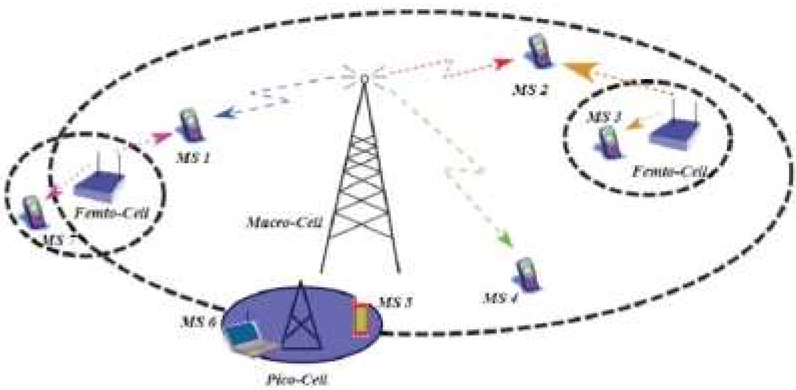


Figure 10: Cell Identification scheme (Liu, Zhang, Yi, Li, & Zhang, 2012)

Angle of Arrival (AOA): determines the positioning of the user by measuring the angle towards the receiver from the transmitter. The transmitters must be capable of calculating such information. This can be done with directional antenna. Yet, the method is unreliable, since it is prone to multi-paths, plus it requires the line of sight to detect the receiver, which us hard to counter in the harsh indoor environment (Lenihan, 2004).

Time of Arrival (TOA): measures the exact distance by using the travel time of the signal from the transmitter to the receiver. Using the equation R = time x speed, where speed is a constant, only time needs to be measured to determine the exact location R. However, in order to get more accurate results, synchronization of the receivers is needed (Lenihan, 2004).

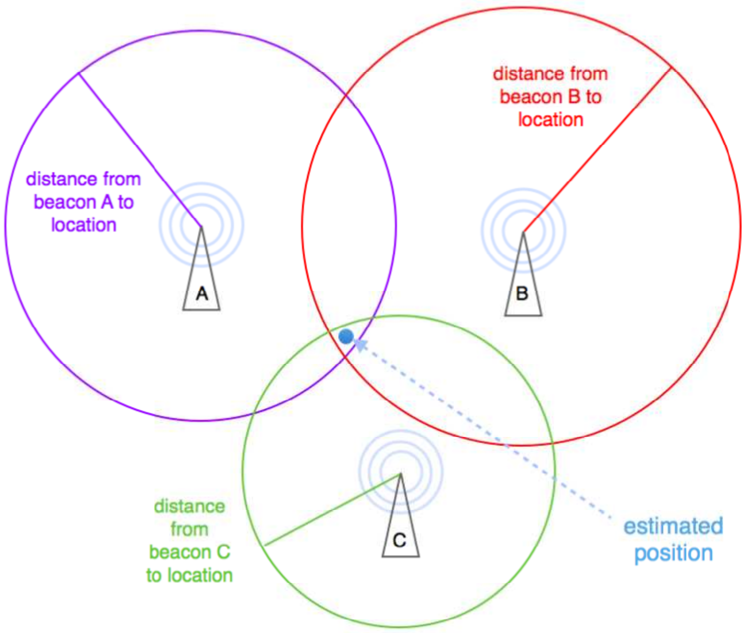


Figure 11: Time of Arrival

Differential Time of Arrival (TDOA): The position is determined from the intersection of the locus of the time difference of arrival at the receiver, which is hyperbolic in a two-dimensional plane and hyperboloid in three-dimensional space. A TDOA system requires a number of base stations that is one greater than the number of dimensions. Accuracy is similar to TOA subjected to the time of arrival measurement and the time synchronization between base stations in the system (Lenihan, 2004).

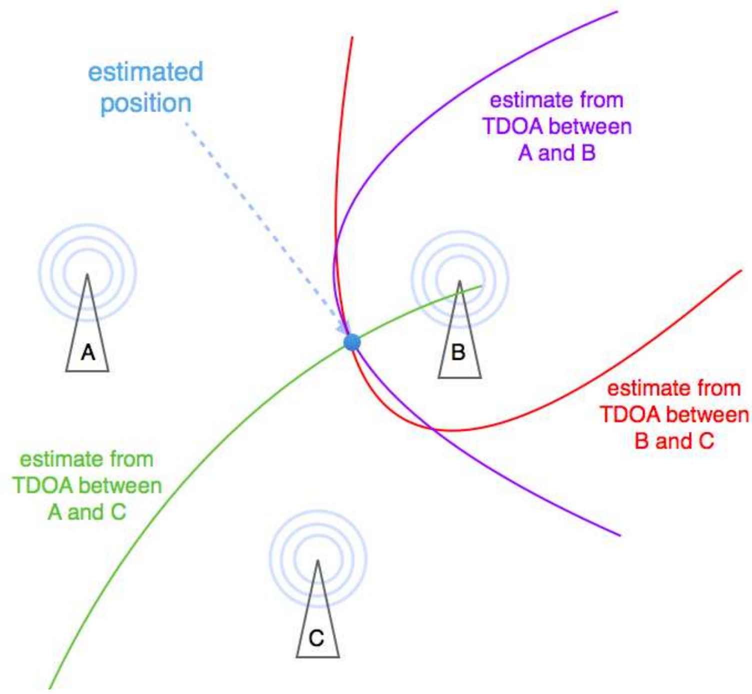


Figure 12: Differential Time of Arrival

Triangulation: In an environment with known propagation losses, signal strength can be directly converted to distance (Lenihan, 2004). In free space, signal strength varies with the inverse of the square of the distance from transmitter to receiver. To accurately convert to distance in a real setting, factors such as antenna gains and interference from objects in the signal path must be accounted for. This method's accuracy depends on the accuracy with which the propagation losses can be estimated.

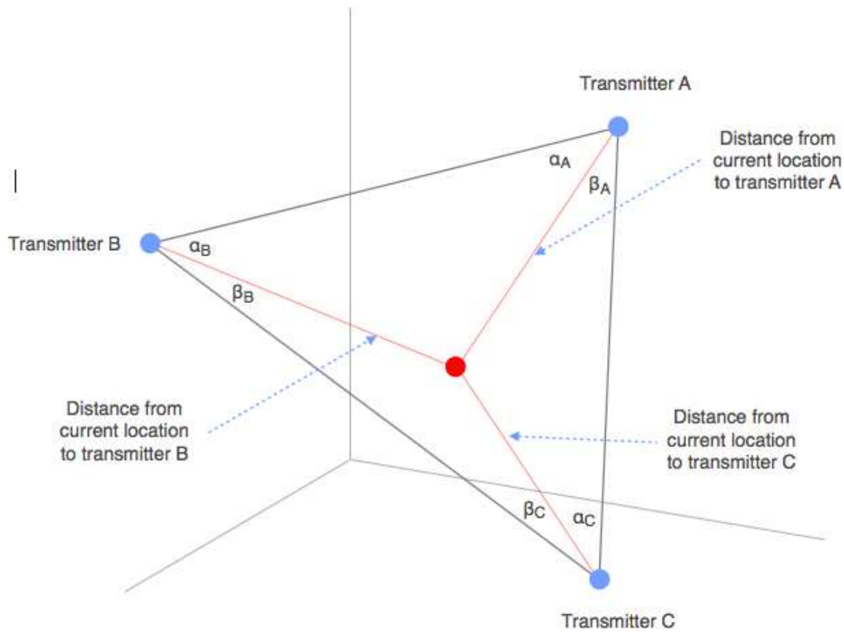


Figure 13: Triangulation

Location fingerprinting: Location fingerprinting is a positioning technique that compares measured RSSI data to a database of expected values to estimate location (Monoj-Kumar-Raja, 2010). Typically, measurements are taken in an arbitrary grid pattern around the building. A multiple matrix correlation algorithm can be used to search this database for the best match, thus giving a position estimate. This method is highly accurate but takes a long time to implement.

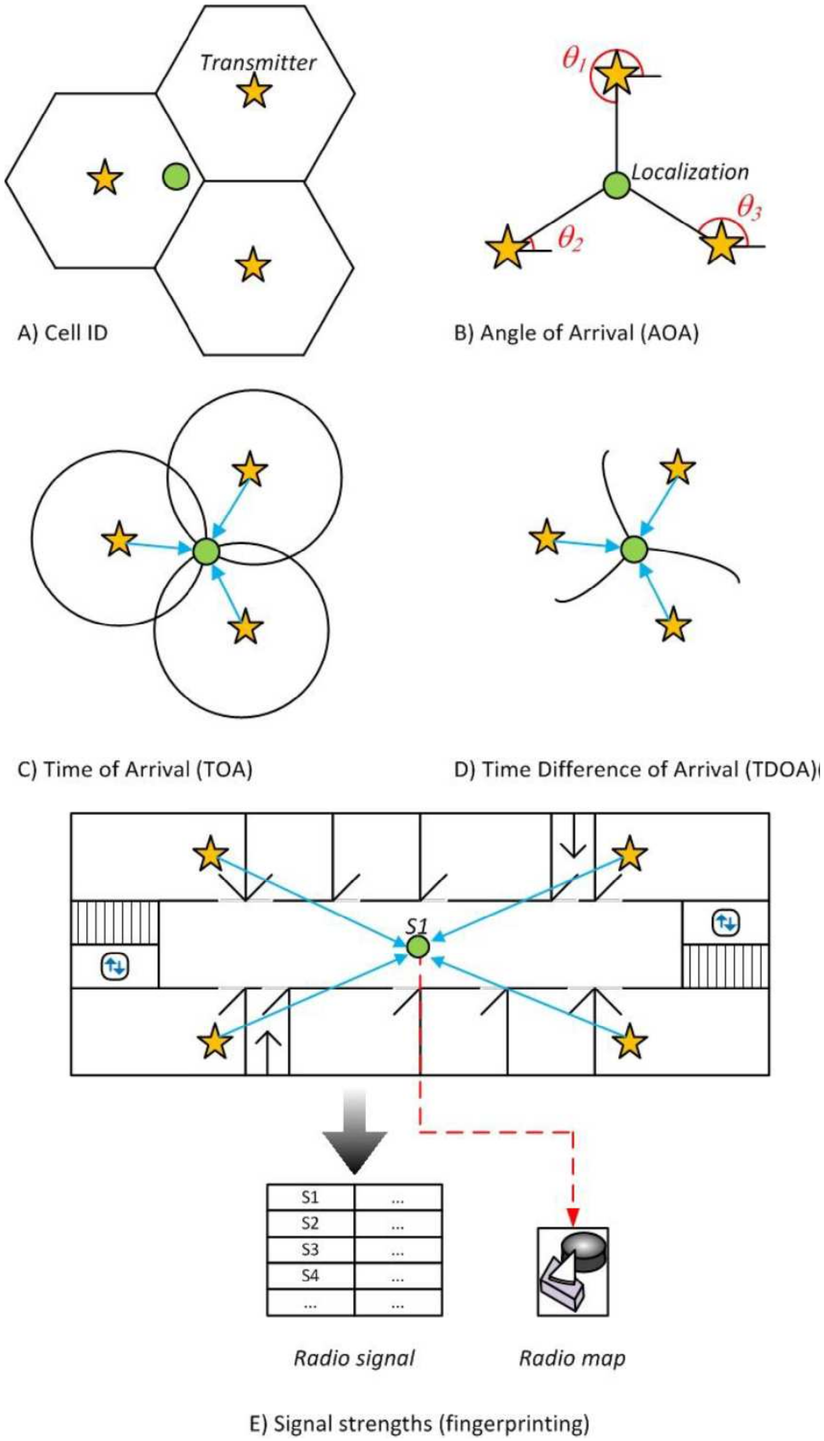


Figure 14: Allocation methods based on *Woo et al. (2011)*

5.3. Indoor wireless positioning system using Smart Phones

The possibilities offered by 3 available resources in smart phones: Wi-Fi radio, cellular communications and accelerometer. The use of Wi-Fi signals as a potential positioning system within buildings has opened doors for many applications. This is because ubiquitous availability of Wi-Fi signals in almost all the buildings, so no additional hardware is required to install a positioning system in the buildings (Monoj-Kumar-Raja, 2010).

Two basics methods of finding the position of the user given the signals strength of Wi-Fi are trilateration and location signature/fingerprinting.

The trilateration method can compute the user location on the fly from the signal strength data and know the positions of the Wi-Fi signals transmitters also called Access Points (APs). The distance of the user from any Wi-Fi transmitter is inversely proportional to the signal strength obtained from that transmitter. One can find the user's location if at least 3 Wi-Fi transmitters are visible.

But this method of trilateration does not work well in the indoor environment because of the signals undergo various interference and multipath problems due to the features like walls, equipments within the building and so the trilateration will not result in correct position of the user. (Li, Salter, Dempster, & Rizos, 2006).

On the other hand, fingerprinting techniques have already proved to be able to deliver better accuracies. The mobile terminal estimates its location trough best matching between the measured radio signals and those corresponding to locations previously registered in the radio map (Martin, Vinyals, Friedland, & Bajcsy, 2010). This technique consists in two stages: Training phase and online phase (Monoj-Kumar-Raja, 2010), (Meng, Xiao, Ni, & Xie, 2011), (Martin, Vinyals, Friedland, & Bajcsy, 2010).

Training phase, also called offline phase, in which radio map of the area is built. RSSI values from different beacons are recorded at different locations; the separation between these chosen locations will depend on the area in the study, and for instance, for indoor environments this separation can be of around a meter (Martin, Vinyals, Friedland, & Bajcsy, 2010). The training data is stored in a database.

The other phase is called Online phase, in which the mobile terminal infers its location through best matching between the radio signals being received and those previously recorded in the radio map. Localizations algorithms employed in this case generally make use of deterministic or probabilistic techniques (Martin, Vinyals, Friedland, & Bajcsy, 2010):

Deterministic techniques store scalar values of average RSSI measurements from the access points. The most relevant techniques in this group are:

a) "Nearest neighbor in signal space".
b) "Nearest neighbor in signal space-average", choosing *k* nearest neighbors and calculating the centroid of that set.

c) "Smallest polygon", selecting several nearest neighbors which will from various polygons and the centroid of the smallest polygon will be considered as the estimated location.
d) "Naïve Bayes method", the given set of signals at the user's location is processed and the probability of the each and every point in the database being the user's location is calculated using the probability density function. Then the point with the highest probability of being the user location is picked and declared as the user's location (Monoj-Kumar-Raja, 2010).

PART II – PROJECT OVERVIEW AND DESIGN

CONTENTS

SUMMARY

This part is about the project overview and the design of the application. First points introduce the whole application: the goal of the application, the mobile platform used and the general design. In the point 12, the server part is explained. The last point is the client part where there are explained the design and the implementation of the mobile application showing different schemas and shot screens to help understand how works and how is designed the application.

6-. GOAL, OBJECTIVES AND SCOPE

The main goal of this system is to facilitate indoor positioning for mobile devices inside the framework of the viscaUJI project of the UJI Smart Campus. This tool will be applicable to the others subprojects of the viscaUJI project (navigation, place-finder…).

The goal of the application is provide to the user an tool to carry out the indoor positioning. The indoor positioning using Wi-Fi data have two main methods: trilateration and fingerprinting. The application provides two main functions to carry out the offline part of these two methods: add antenna and add Wi-Fi fingerprint point.

The system will be easy to implement in buildings that have existing wireless connectivity. The system will be tested first of one in the INIT building (TI) to test the system and methods, and then it will be incorporated the others buildings of the UJI campus.

To function as intended, the system must meet these primary objectives:

- ✓ The application must be used to carry out the first phases of the method Wi-Fi Indoor Positioning System (WIPS).
- ✓ The application must be register all needed data for WIPS and must be store the data in a server to use in right way this data later.
- ✓ The device must have an intuitive user interface.
- ✓ The mobile application must to be compatible with the requirements of the viscaUJI project and use its data to visualize the information.
- ✓ This system must be a tool for the others applications or projects of the Smart Campus project.

The scope of the project is for one hand the Wi-Fi location technology and its application in an indoor location based service implemented and designed for mobile devices. By the other hand the application of this system and integrating in the viscaUJI project of the UJI Smart Campus.

The ambit of the project is limited to Wi-Fi localization technology and its application in an indoor location based service. To implement the geographical part of application and the architecture of the geodatabase the scope will be the ESRI technology, using ArcGIS server, ArcGIS SDK for Android and others ESRI APIs.

The project will be developed, initially, at the TI building of the UJI. Then, when the system and the prototype provide the indoor location with some precision, the system will be extended for all UJI buildings.

7-. DESIGN REQUIREMENTS

The design includes five subsystems. These subsystems and their responsibilities are summarized in the next table:

Subsystem	Responsibilities
Geodatabase	Directory of geographical information that contains the UJI campus data and will contain the registered data.
User Interface	Allow the user all provided functionality.
Cartographic Basemap	Is the design of the visualized data provided by the viscaUJI geodatabase
Registering data	Allow the user to do the offline part of the Wi-Fi Indoor Positioning with the fingerprinting method and allow to the user to insert the position of each antenna Wi-Fi.
Positioning	Locate the user in the building.

Table 2 The design requirements include five subsystems and their descriptions

Each subsystem has its own specific requirements. In the ***geodatabase subsystem*** there are two geodatabases used in the project. One is the geodatabase of the smart campus and the other aims to store and manage the data to perform the indoor positioning. In the first database the application only can perform queries about these geographic and alphanumeric data order to show, visualize and interact with the UJI cartography. The other database has a correct structure to store Wi-Fi data (Wi-Fi fingerprints) related with its georeferenced point. The data stored is used to do different analysis focused on the Indoor Positioning.

The ***user interface subsystem*** provides to the user on one hand several tools to interact and visualize the geographic data of the campus, and on the other hand two functions one for add antennas and the other for add a fingerprint measurement. This User Interface (UI) is very intuitive and fully exploited the size of the mobile screen. The UI provide same alerts to inform the user if one is not using correctly the tools. The UI take into account the design of the whole project viscaUJI and the colors and the final aspect is coherent respecting the viscaUJI guidelines.

The ***cartographic basemap subsystem*** is the organization or the structure of the campus geographical data in the mobile application. There are same features and layers in the viscaUJI geodatabase of the campus. These layers are used as dynamics maps organized in the case of the building interior layers by floor. The user can select the UJI basemap (color map or grey scale map), and choose visualizing between the building layer and building interiors layer. When one choose the building interior layer the floor 0 is loaded by default but user can change the floor.

The ***registering data subsystem*** consists in two functions that are implemented in the mobile applications. Their functionality is register data of the Wi-Fi fingerprints and data of the antenna. This data is stored in the server following the architecture of the Wi-Fi indoor geodatabase. These functions register geographic data of the points and some attributes depending if the points are Wi-Fi scans or antennas (floor, map coordinates, antenna name, Wi-Fi Access Points registered…).

The aim of the ***positioning subsystem*** is locate the user in the building. This subsystem register the fingerprint at the point where is the user and compare its fingerprint with the fingerprint database. This subsystem is the second part of the Wi-Fi fingerprint method to positioning the device, i.e. the online phase.

8-. DESIGN

The system consists of five sub-systems as illustrated in the next figure. The whole system is divided in two parts, the client part (a mobile device) and a server part. These parts are related between them and the subsystem inside the parts too, as shown in the figure. The figure is a general scheme of the system.

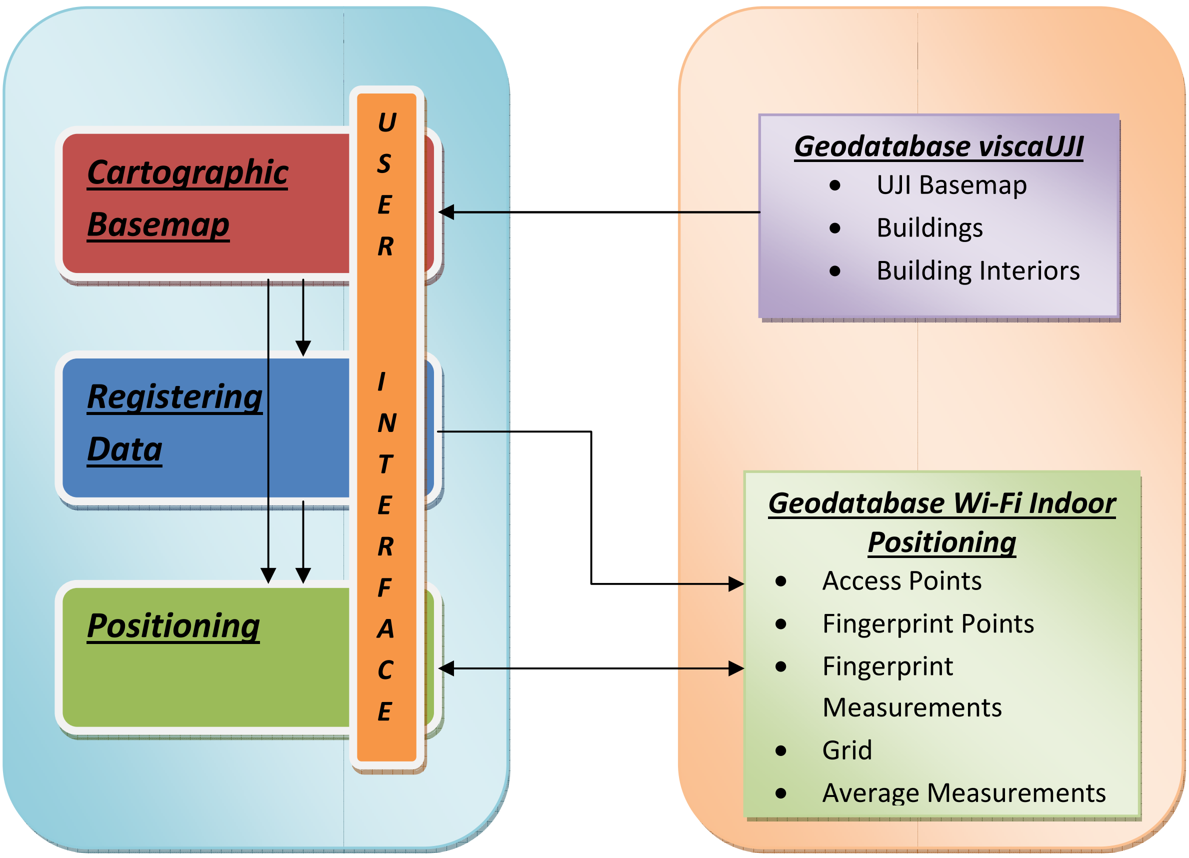


Figure 15: General schema of the system designed

The geodatabase viscaUJI provide to the system the basemaps: two basemaps of the UJI, a map of buildings and a map with different layers, one per floor. The client, (the cartographic subsystem) manage and organize these information using the tools provided by the ArcGIS SDK. This subsystem provides same tools to allow the user to visualize what it wants.

There are two main functions of the registering subsystem. The first one is register an Access Point or antenna and the other is registering a Wi-Fi fingerprint. At the moment this function of register a fingerprint is focused to add a fingerprint in the geodatabase (perform the offline phase of the Indoor Positioning method).

Add fingerprint register function uses the data of the Wi-Fi Manager of the mobile phone. Then, when the user touch on the mobile screen and the system register at this point the fingerprint data and save in the Wi-Fi Indoor Geodatabase the point, the fingerprint measurement and the floor that the user is displaying. This function store the points data in the fingerprint points feature layer and the measurements in a table that are in the server.

Add Access Point or antenna function add an antenna where the position that the user touch on the screen. The informatics department of the UJI campus provided us information of the antennas: name or code and building and floor where each antenna is. This information is included in the system and when the user touch the map, a dialog appears and the user select the name of the antenna. Then the user select the name of the antenna that it want to add. If this antenna is in the same floor that the user is viewing the antenna is added, but if not, the system don't allow to the user to add this antenna.

The positioning subsystem use many of the methods of the add fingerprint functionality. The difference is that this fingerprint is not added to the geodatabase. This subsystem compares the current fingerprint with the geodatabase, and returns the point that has the most similar fingerprint previously registered in the offline phase.

9-. MOBILE PLATFORM

The mobile platform selected to develop the application is Android. Android has different versions of their platform. There are same many active devices running with different versions of the Android platform. In the next graphic and table one can observes the relative number of devices compared with the version that are running.

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Eclair	7	2.4%
2.2	Froyo	8	9.0%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	47.4%
3.1	Honeycomb	12	0.4%
3.2		13	1.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	29.1%
4.1	Jelly Bean	16	9.0%
4.2		17	1.2%

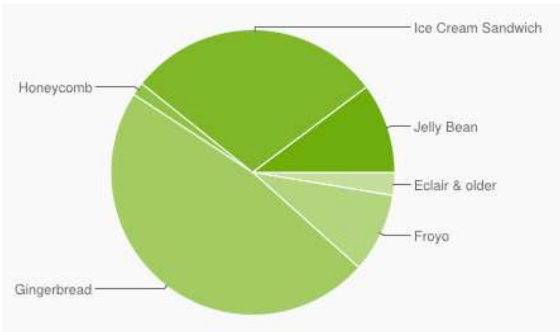


Table 3: Comparison between the Android platform version respect the relative number of devices running with this platform (Android developer)

Higher versions of the platform has new tools and functionalities, and there are same improvements respect the older versions. To select the target version of the platform one has to take into account two main factors: the most recent as possible and the percentage of devices running with this versions. The higher versions are completely compatible with the previous.

Each version of the Android platform has a specific API to develop the applications. In this work the target version selected to develop the application is the ***2.3.3***, named ***Gingerbread***. The API used to develop an application with this version is the API number ***10***.

In next table one can observe the selected version and the relative number of devices that can use the developed application.

Version	Codename	API	Distribution
2.3.3	Gingerbread	10	88,2%

Table 4: Android platform version used and relative number of devices compatible with this version (*31-1-2013*)

10-. POSITIONING TECHNIQUE

The method that perform the application developed is the Wi-Fi fingerprint technique. This technique is explained in the chapter 5.3.

There are two phases in this technique: offline phase and online phase. The application developed performs correctly the first phase, the offline phase, called mapping phase.

To carry out the mapping phase, the application provides a function to register the fingerprints. Three points have been stored by each cell of the 3x3 meters grid. The average of these fingerprints are calculated and assigned for each cell. This operation is performed using ArcGIS tools and a new table of these averages are created and shared in the server.

11-. SERVER PART

The data used to visualize the UJI campus are in two servers. These servers are located on the campus. One of these servers contains all right data of the Smart Campus (Geotec server) and the other one contains all testing data (MasterGeotec server).

Whoever that want to have access to these servers will have to be connected to the UJI net. There are two options to connect: directly connected to the UJI net or through a VPN connection.

In these servers there are installed ArcGIS Server. ArcGIS Server provides a platform for sharing GIS resources (maps, feature layers...). One can access to this resources accessing trough Internet. Depending the settings of the resources, one can access, query this information, apply edits, delete,... The main advantages of use a GIS server is that the GIS resources are the same as sharing any data trough any kind of server technology: the data is centrally managed, supports multiple users and provide clients with the most up-to-date information.

The ArcGIS Server REST API (Representational State Transfer), provides a simple, open Web interface to services hosted by ArcGIS Server. All resources and operations exposed by the REST API are accessible through a hierarchy of endpoints or URLs for each GIS service published with ArcGIS Server (ESRI).

The ArcGIS REST system is a hierarchy of resources. Some resources are in and by themselves (catalog, map, layer, etc.). Other resources are produced as a result of an operation (exporting a map results in a map image resource of an operation).

One can observe in the following figures the interface of the REST API through a browser using the URL of the servers used in this work:

<u>**Geotec server**</u>: *http://geotec.dlsi.uji.es/arcgis/rest/services/UJI*

Figure 16: Service directory of the Geotec server

<u>**MasterGeotec server**</u> : (in this case the folder of this project) *http://mastergeotech.dlsi.uji.es/arcgis/rest/services/WifiIndoor*

Figure 17: MasterGeotec server, inside the folder of this project

The resources used at the mobile application are the ***feature services*** and the ***map service***. The map services are used to visualize information. In some cases are used as a dynamic map and others as a tiled map. The dynamic map visualization offers a dynamic interaction between the labels of the map depending of the scale of visualization. The feature services are used to register the data using the mobile device. Feature services support the "Apply edits" operation. Therefore this is the best way to populate a geodatabase of this system.

The following list show all data used as a map in the application.

Services used that are in the Geotech server:

- UJI/Buildings (MapServer)
- UJI/viscaUJI desaturado (MapServer)
- UJI/viscaUJI (MapServer)

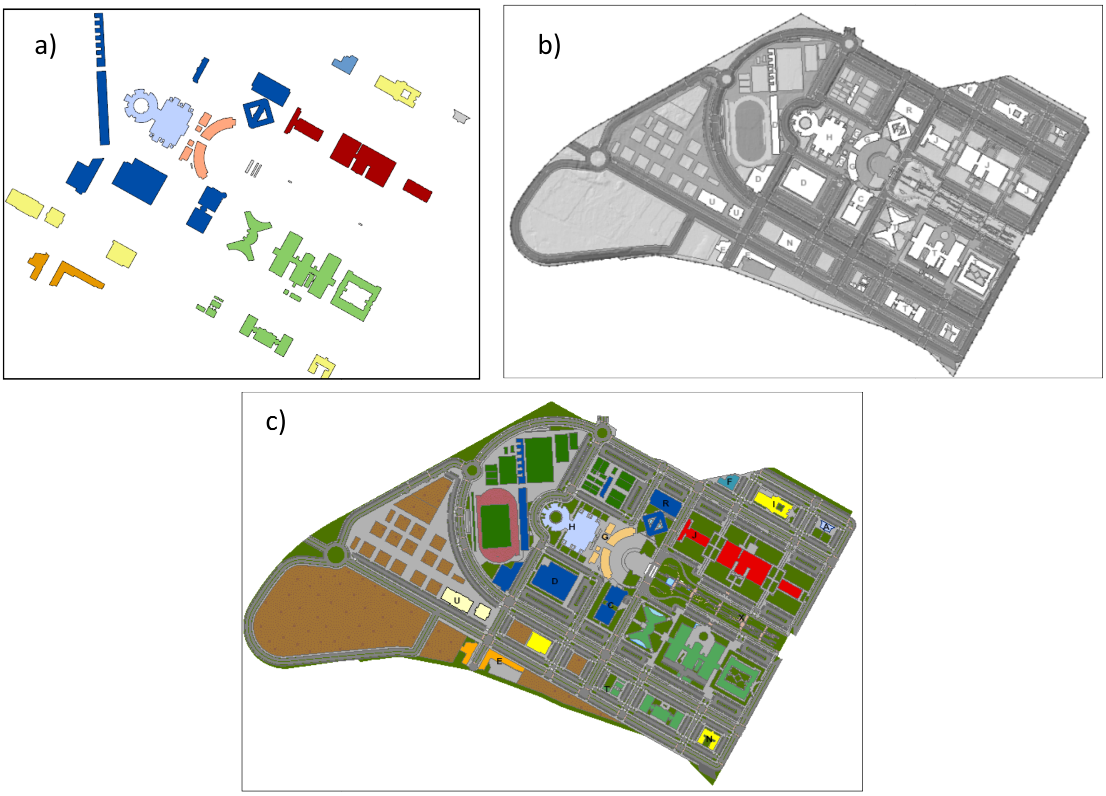


Figure 18: a) UJIBuildings; b) viscaUJI desaturado; c) viscaUJI

Services used that are in the MasterGeotech server:

- WifiIndoor/Antenna (FeatureServer)
- WifiIndoor/WiFi ScanPoints (FeatureServer)
- WifiIndoor/BuildingInteriorbyFloor (MapServer)
- WifiIndoor/Grid (MapServer)

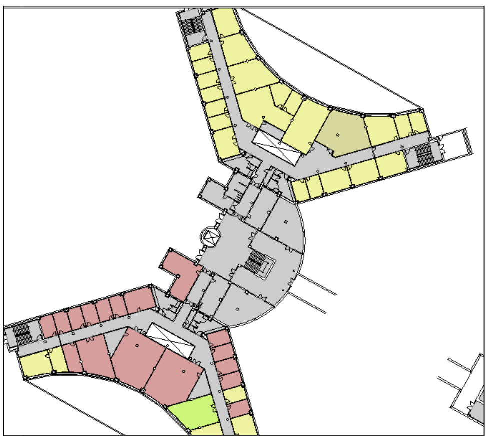


Figure 19: UJI Buildings interior by floor

The building interior by floor map is based on the data of the viscaUJI geodatabase, but the original data contains all data of the building in the same layer but each element in each floor has the attribute floor that specifies the floor where the data is. *BuildingInteriorbyFloor* has different layers, one per floor. In this way the data is more light to use in a mobile device.

The Antenna feature contains the Access Points and its geometry type is point. In this feature the information registered is the Name of the Access Point, the Floor and the MAC. The WiFi ScanPoints feature contains two layers: WiFi ScanPoints and Measurements. This feature contains the data of the fingerprint database. In the WiFi ScanPoints layer have the information of the points: Floor, coordinates GPS provided by the device using the coordinates provided by the network and the ID of each point. The Measurements layer is a table that contains the measurements of the WiFi signals measured in each point. The fields of this layer are: ID of the point, the name of the network, the MAC of the Access Point and the intensity received.

The next table shows the fields of this features:

Feature Name	Layer	Fields					
Antenna	Antenna	OBJECTID	FLOOR	NAME	MAC		
Grid	Grid	OBJECTID	FLOOR	ID_CELL			
WiFi ScanPoints	WiFi ScanPoints	OBJECTID	ID_POINT	FLOOR	X_GPS	Y_GPS	
	Measurements	OBJECTID	ID_POINT	FLOOR	MAC	NAME	INTENSITY
	Average Measure.	OBJECTID	ID_CELL	FLOOR	MAC	NAME	INTENSITY

Table 5: Features fields of the WiFi Indoor Geodatabase

Next figure shows the Class Diagram of this Features.

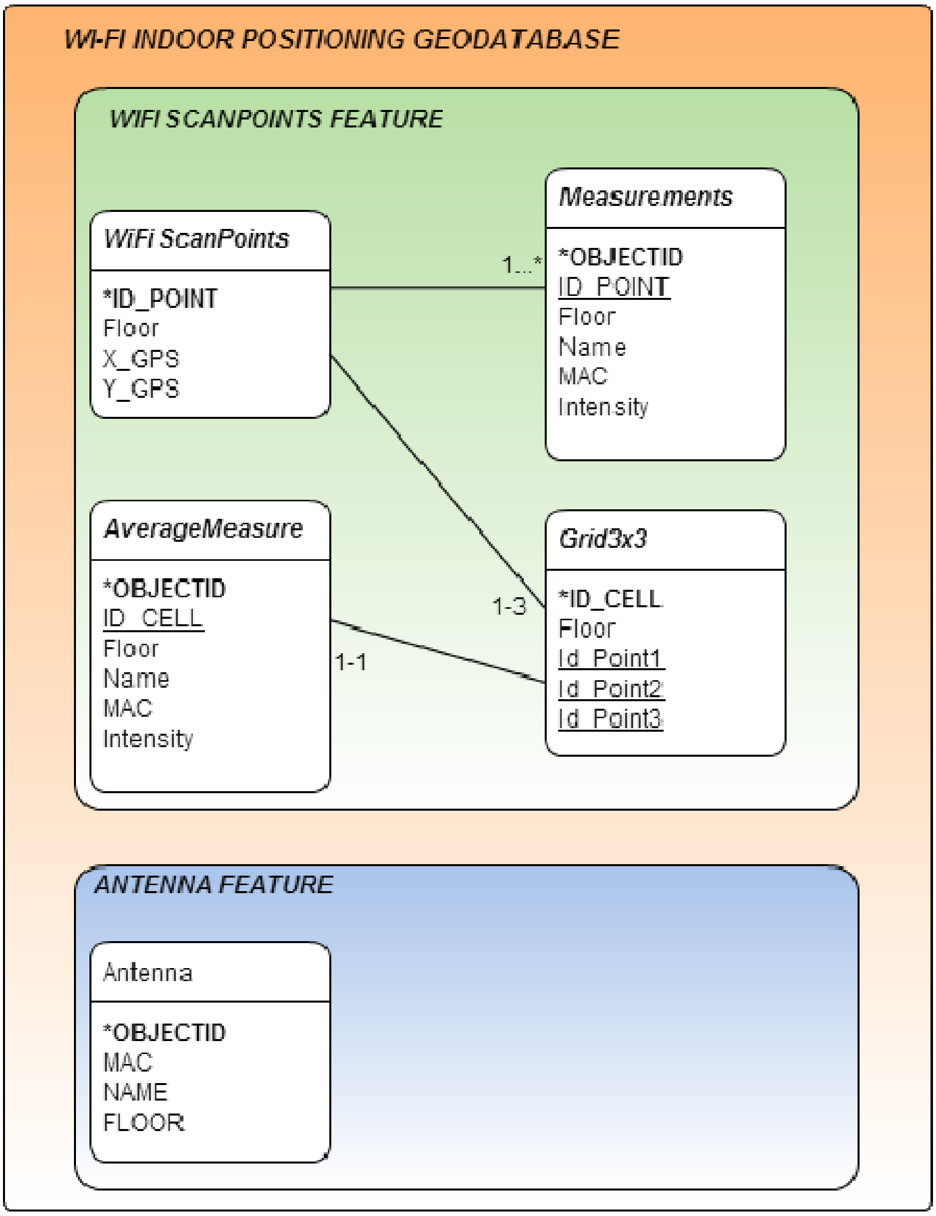


Figure 20: Diagram of the Geodatabase Architecture

12-. CLIENT PART

12.1. Software System Design

To develop the application different tools are needed:

- Eclipse IDE for Java Developers
- Android SDK
- ArcGIS SDK for Android

Eclipse is the software used to develop the application. In this applications the two plug-ins (Android SDK and ArcGIS SDK) have been installed. This two SDK provide the necessary tools and libraries to target the application to the Android platform with whole ArcGIS functionality.

The development of an Android application is carried out by developing the successive screens that comprise the application. Each screen is an interface because it will be contain images, buttons, text, etc. In Android, each of these interfaces are known as Activities.

An Activity is an application component that provides a screen with which users can interact in order to do something. Each activity is given a window in which to draw its user interface. The window typically fills the screen, but may be smaller than the screen and float on top of other windows.

An application usually consists of multiple activities that are loosely bound to each other. Typically, one activity in an application is specified as the "main" activity, which is presented to the user when launching the application for the first time. Each activity can then start another activity in order to perform different actions.

In the next figure there are represented the loops and the paths an activity take between states.

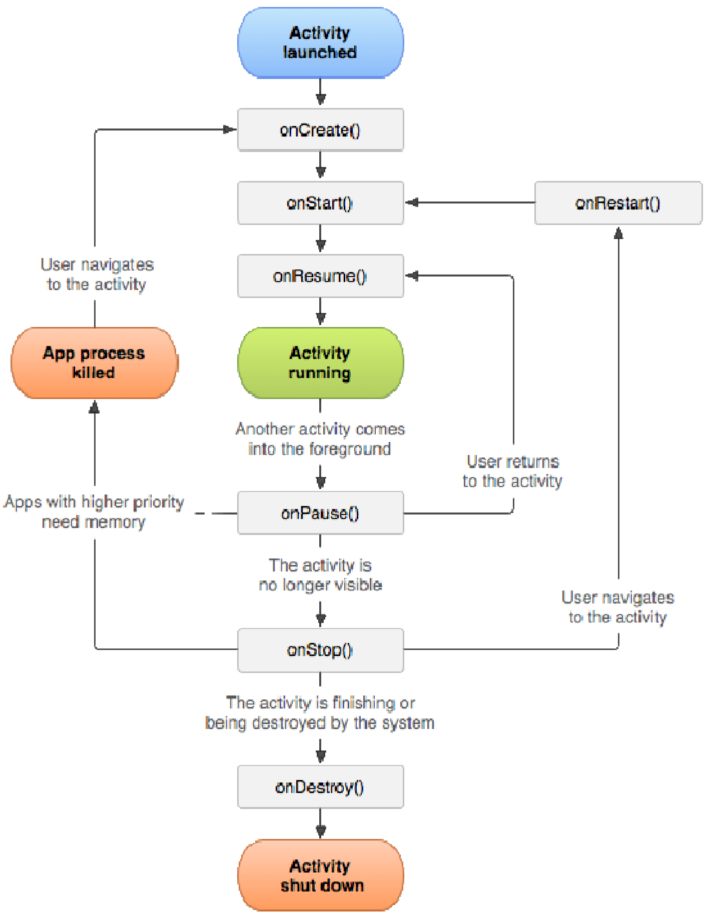


Figure 21: Activity lifecycle (Android developer)

In the application developed there are two activities implemented: the ***main activity*** and the ***indoorpositioning activity***.

The Use Case Diagram of the application is shown in next figure:

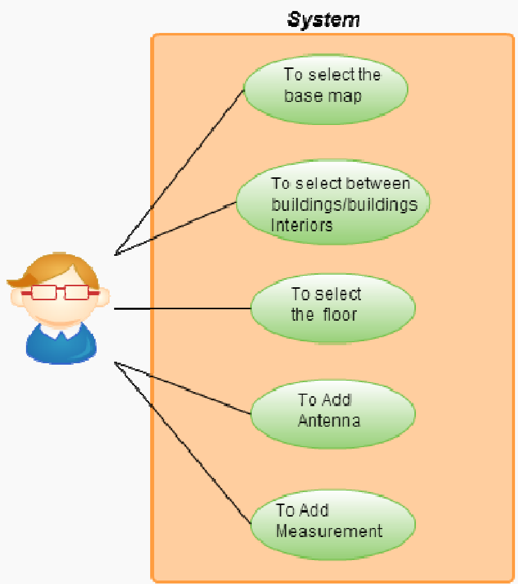


Figure 22:Use Case Diagram

The main activity is a screen that consists on different images about the project (the banner and logo of the viscaUJI and a image of a UJI campus). When the user click on the campus image, the indoorpositioning activity starts.

In the indoripositioning activity are all methods and all functionalities of the this mobile application. Next figure shows the scheme of the activity.

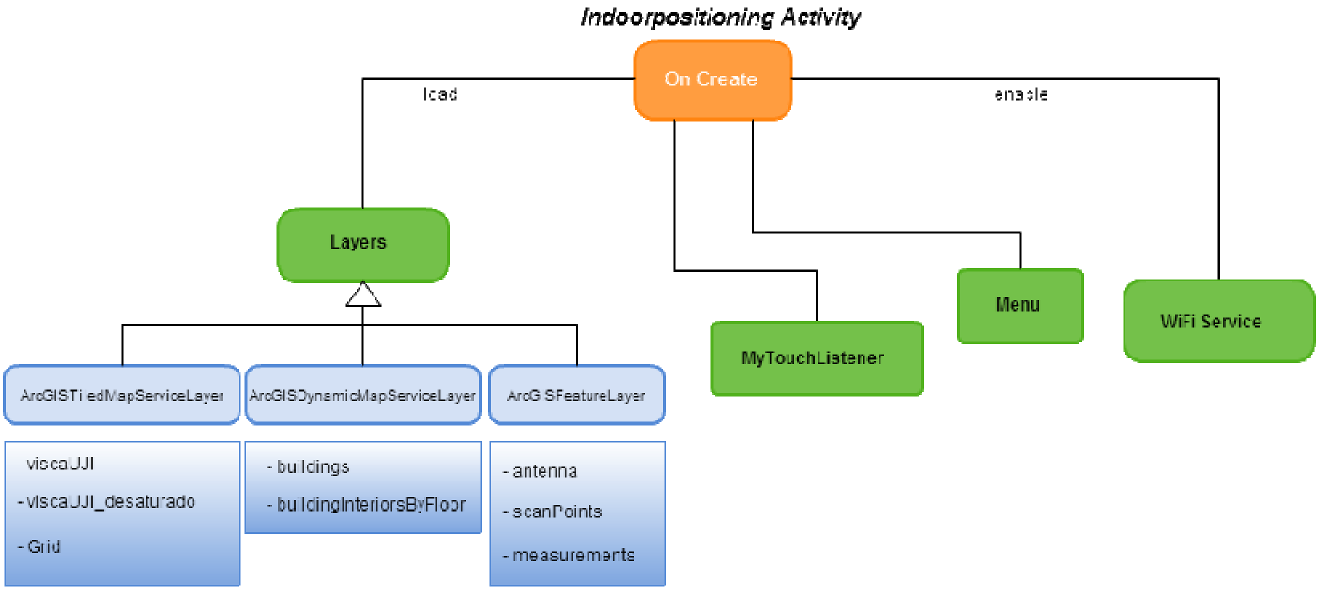


Figure 23:Scheme of the class IndorPositioning

There are five main actions inside this activity: load the layers, enable the Wi-Fi services, initialize the Location Manager, the menu options and single tap on the screen.

Load Layers is the first action that is perform when the indoorpositioning activity is created. The data loaded is type Layer but there are loaded three subclasses of Layers loaded: ArcGISTiledMapServiceLayer (tiled maps), ArcGISDynamicMapServiceLayer (dynamic maps) and ArcGISFeatureLayer (feature layer). Tiled maps and dynamic maps have been loaded as map services from server. This way to load is more light to the mobile device and one can interact with data faster. The map services are used only to show and visualize information, not for edit. The features layers are loaded as features services from the server too but this kind of service allows edit data.

When this activity is loaded there are some data that is not visible, but users can choose what data it will be visible in the app.

Layer name	Is visible?
viscaUJI	*True*
viscaUJI_desaturado	*False*
UJIBuildings	*True*
UJIBuildingsInteriors	*False*
Antenna	*False*
ScanPoints	*False*
Grid	*False*

Table 6: Initial visualization of the layers

Enable Wi-Fi has been performed automatically when this activity is created. This application has permission to enable the Wi-Fi

There are four options in the menu of the application. These options are:

- "Select basemap"
- "Add Building Interiors Layer"
- "Select floor"
- "About"

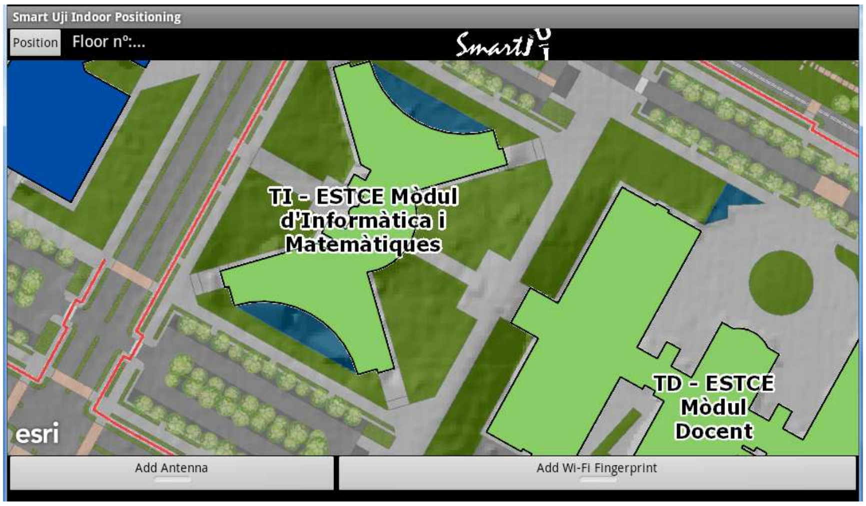


Figure 24: Main screen of the application, displayed by an emulator

Figure 25: Application menu with the options: "Select BaseMap", "Add Buildings Interiors Layer", "Select Floor" and "About"

- "Select basemap" choose between the visualization of the "viscaUJI" or "viscaUJI_desaturado".

Figure 26: Select BaseMap dialog

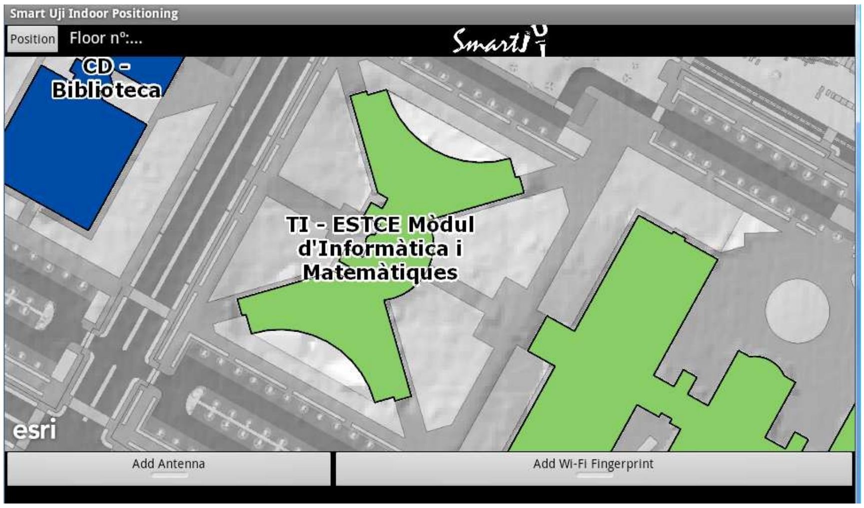


Figure 27: Application displaying the Grey Map

- "Add BuildingInteriors Layer" choose between the visualization of these two layers·"Building Interiors" or "Buildings". When user select the building interiors item the floor 0 is loaded by default.

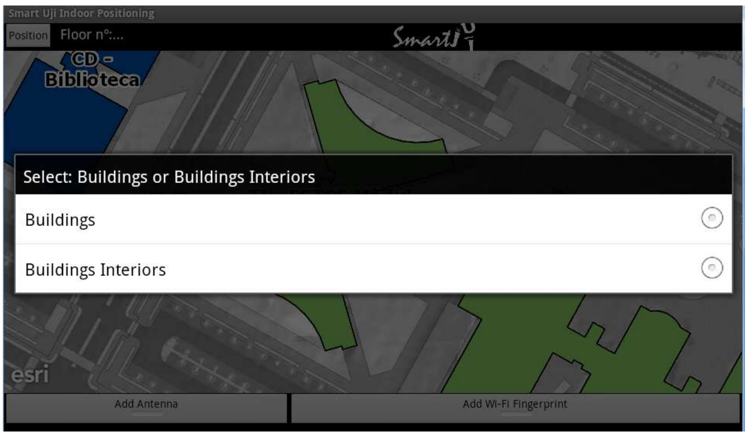


Figure 28: Select Building Interior Layers dialog

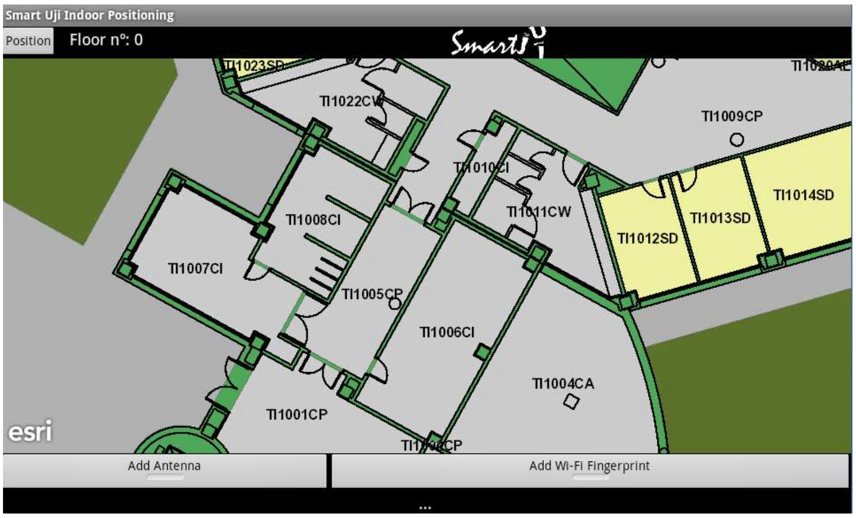


Figure 29: Displaying the building Interiors, floor 0

- "Select floor" option show a list of the number or name of the floors, and user select the floor. This map is dynamic and depending of the scale of visualization different elements will appears.

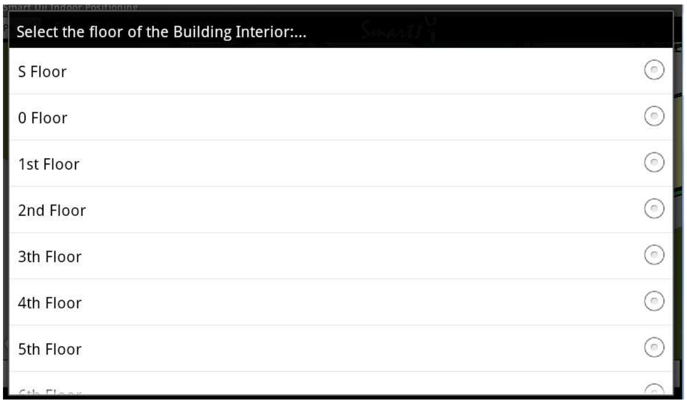


Figure 30: Select Floor dialog

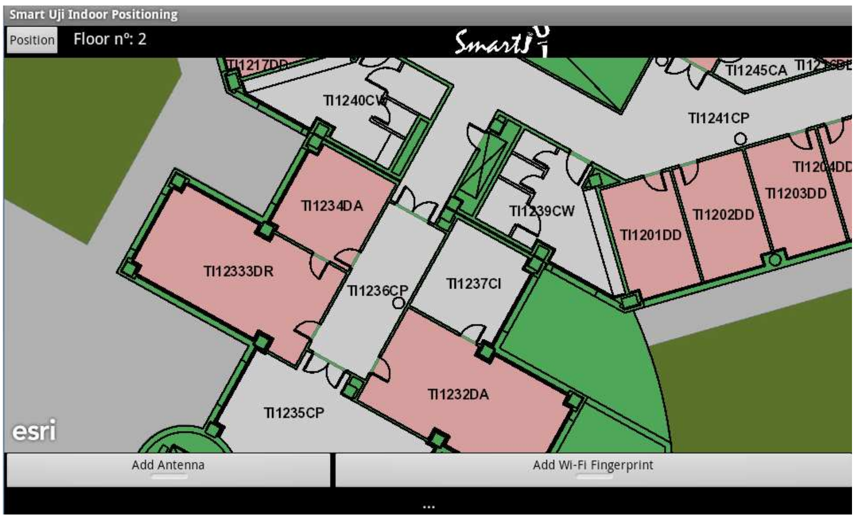


Figure 31: Displaying of the second floor

➢ "About" option shows application and project information.

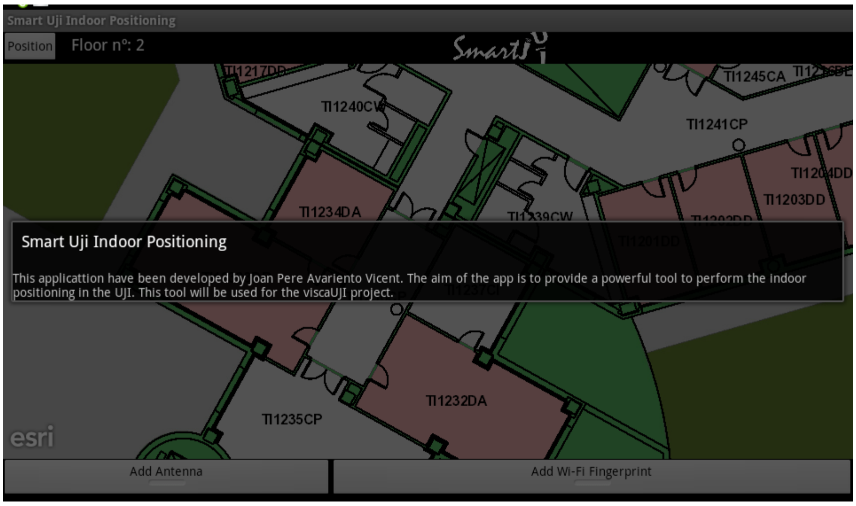


Figure 32: About dialog

In the user interface of this activity there are two toggle buttons. One is "Add Antenna" and the other is "Add Wi-Fi Fingerprint". These functions able the touch listener of the map. Next figure explains the main scheme of this functions.

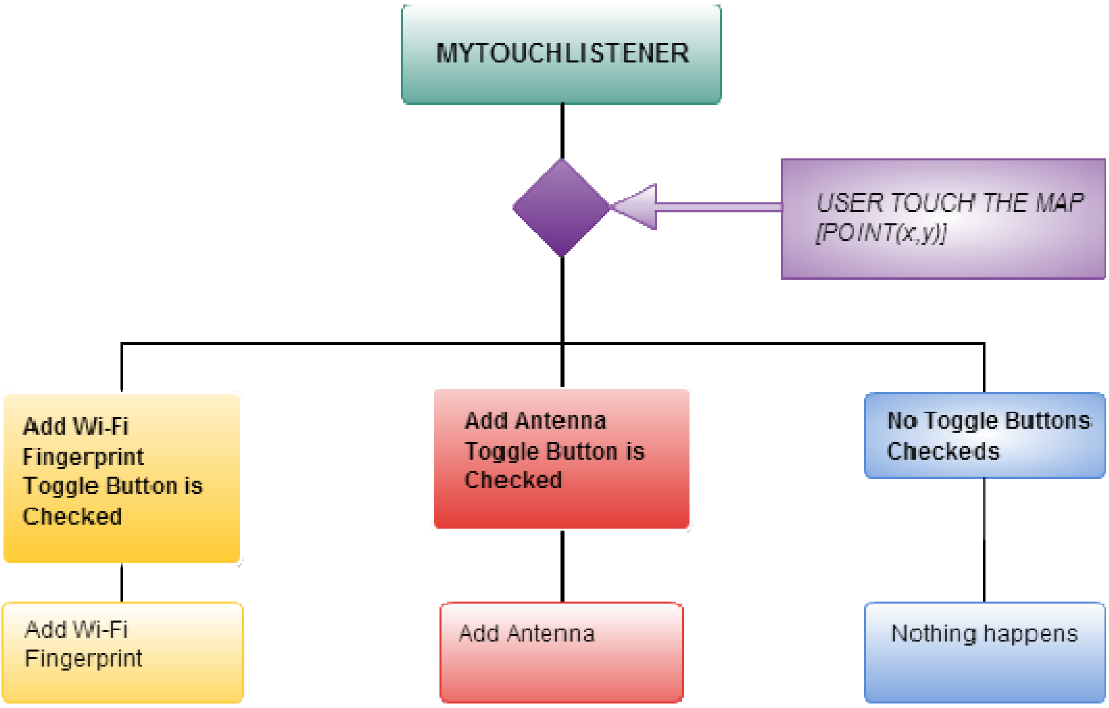


Figure 33: "MyTouchListener" main schema of the application

When the user touch the map, depending of the toggle button checked the result will be different. There are three cases. Next one is "No toggle buttons are checked", then nothing happens. The other one is "Add Antenna" toggle button is checked, then the function of Add Antenna is performed. This function is explained in the next figure. The last case is when the Add Wi-Fi Fingerprint toggle button is checked. This event throws the function Add Wi-Fi fingerprint, explained in the figure ______.

When a user touch the map and any toggle button is checked, then the point when the user have been touched the map is used with the functions to add in this position a point.

Next figure shows the scheme of the Add Antenna function.

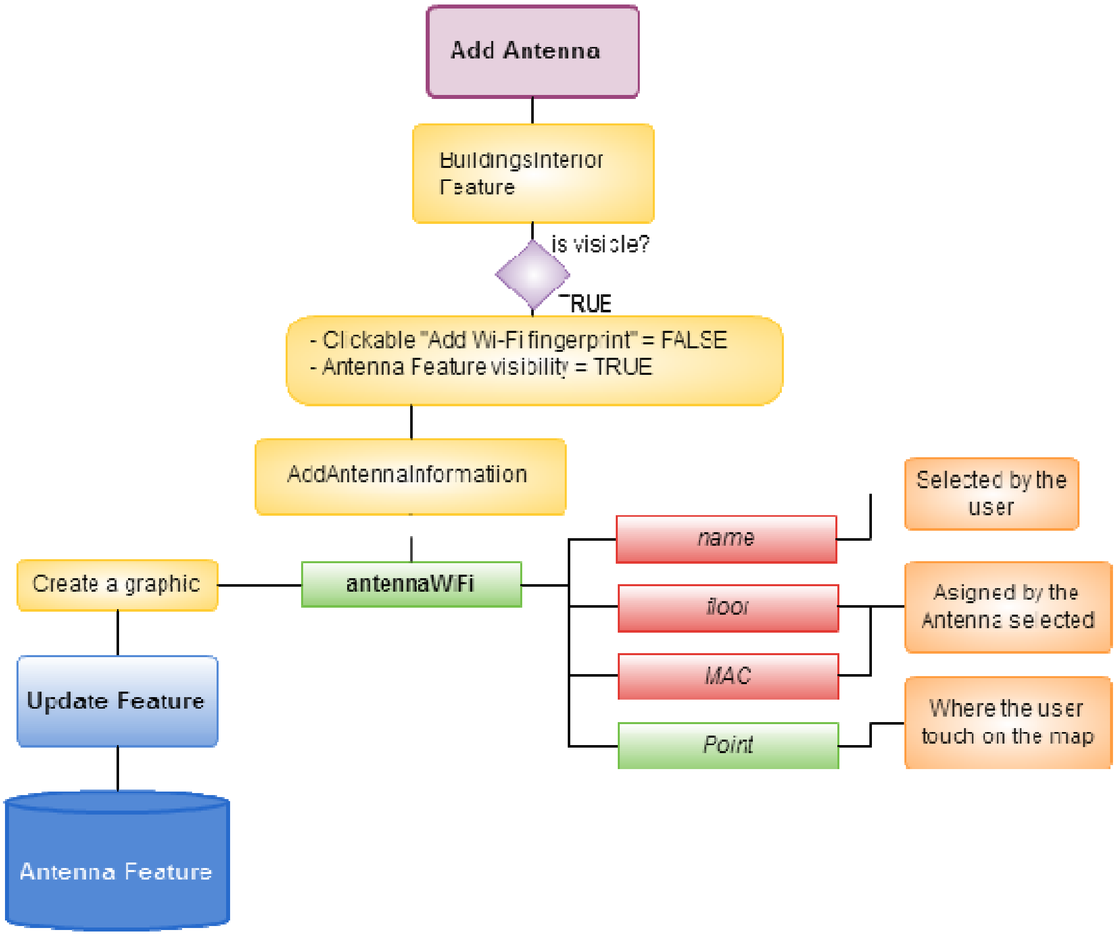


Figure 34: Scheme of the Add Antenna function

Before adding the antenna, the application check if the Buildings Interiors layer is visible. If it is not visible, the application shows a message saying that the user have to visualize this layer. If the layer is visible, then the antenna layer becomes visible.

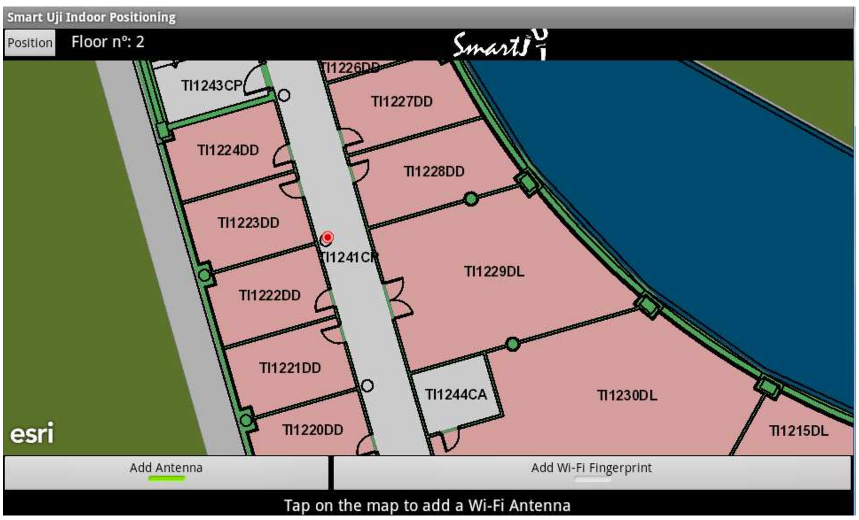


Figure 35: Antenna Layer becomes visible in the application

A list appears in the screen with a list of a names of all antennas. The user choose one, and the attributes of this antenna have stored in an antennaWiFi class. This class contains the name of the antenna, the MAC, the floor of this antenna and the point where the use have been touched on the map.

Figure 36: List of antenna names in the application

The last step of this function is store this antenna. A graphic is created with these attributes and the Antenna Feature is updated with this graphic.

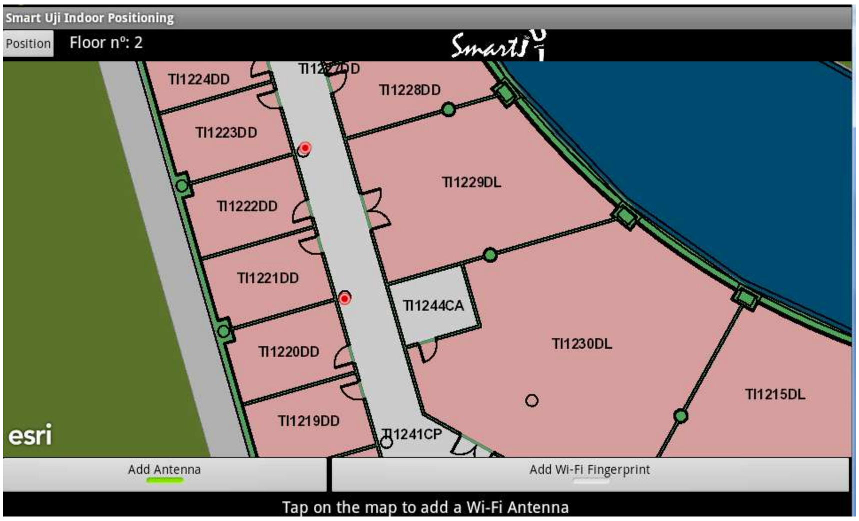


Figure 37: A new graphic appears where the user have touched on the map

Next figure shows the scheme of the Add Wi-Fi fingerprint function.

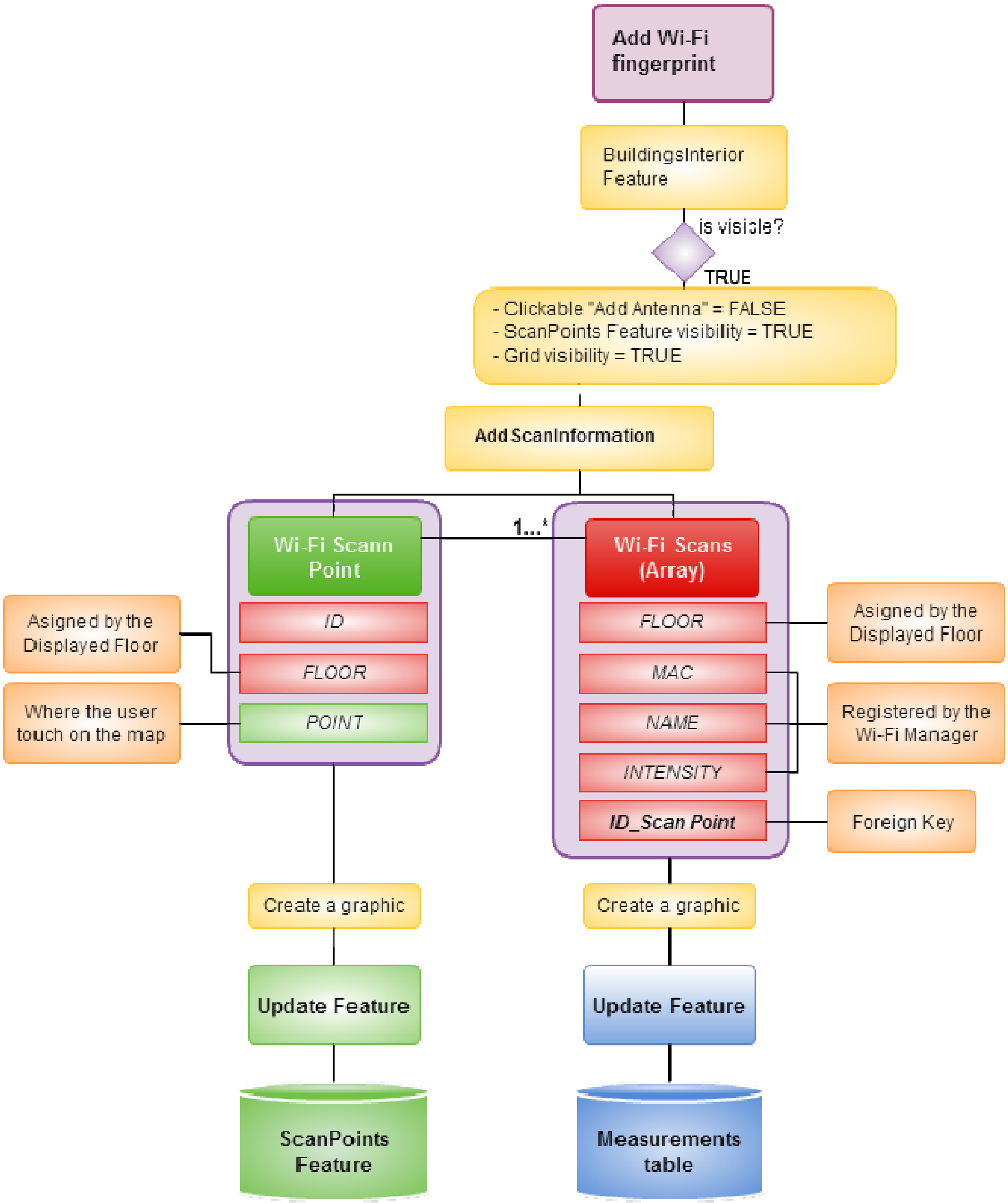


Figure 38: Main scheme of the Add Wi-Fi Fingerprint function

Before adding a fingerprint, the application check if the Building Interiors layer is visible. If it is visible, the scan points layer and the grid layer becomes visible. Then the WiFi manager store the Wi-Fi Access Points (WAP) in an array. Each WAP contains the MAC, the Name of the network, the intensity of the signal. A Wi-Fi Scan Point is created. This object have an unique identifier, the floor where this point is measured and the point where the user have been touched on the map. A graphic is created with this attributes and the ScanPoints Feature is update with this point.

Figure 39: Wi-Fi Scan Points measureds by the user

By other hand the array with the WAPs is stored in the Measurements table. Each row of this table contains one WAP registered by the Wi-Fi manager and in this row is stored as an attribute the identifier of the point where the Fingerprint is measured.

12.2. Registering Fingerprints

There are two steps to carry out the mapping phase. The first step is mapping the Wi-Fi fingerprints and the other is calculate the average of these fingerprints.

Mapping the Wi-Fi fingerprints registers all WAPs that the device find. Then this measurements are assigned to a point. The application shows a 3 x 3 meters grid. User that are performing this step has to take three measurements at least by each cell.

The second step is perform the average of the intensity measured per each cell. Each cell has at least three points measured. Each point has different WAPs measured. Per each cell the average of the intensity of the each WAP is obtained and assigned to the cell centroid.

The second tool is performed using ArcGIS desktop. The Scanned points feature and the Grid feature have been intersected. Working with the tables of attributes the average is calculated.

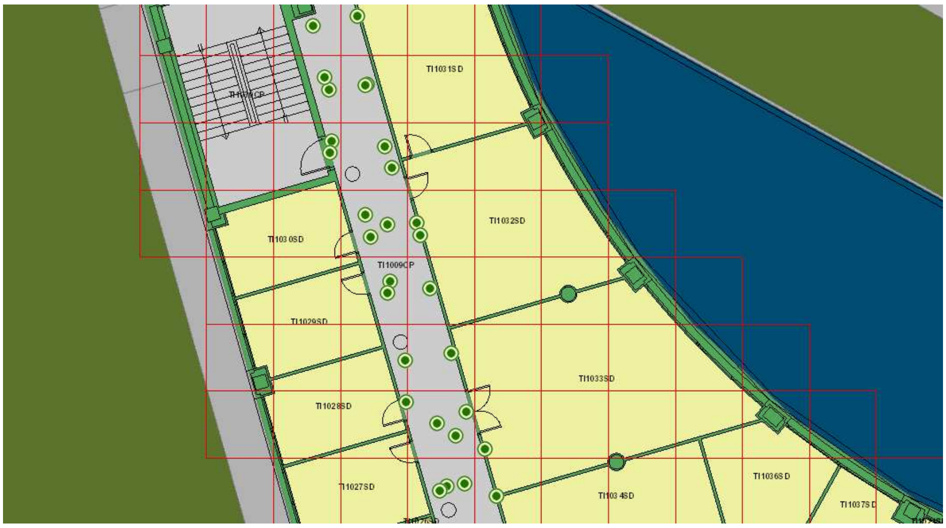


Figure 40: Visualization of the Scan Points and the grid through ArcMap

13-. RESULTS

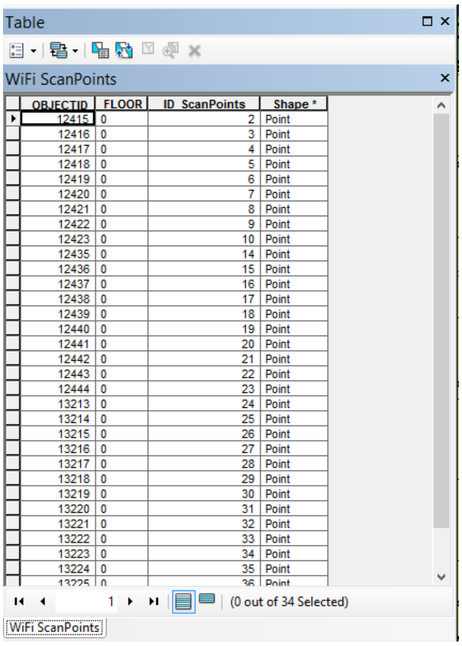
The main result of this work is the application developed. This application is a tool to carry out the mapping phase of the Wi-Fi Indoor Positioning. By one hand the application provide a powerful tool to navigate with the UJI cartography, showing the buildings, the name of these buildings, the buildings interiors, the name of the labs and classroom,... By the other hand the application provide a tool to measure and store the Wi-Fi Fingerprints. These measurements are stored in a geodatabase that relates the position of the measurement with the data of the measurements.

Figure 41: Different interfaces of the training phase of the application. Visualization of theUJI buildings and building interiors

In other way, the results are the fingerprints stored in this database. This cloud of points are the database to in the future perform the Indoor Positioning.

The database that it will use to the Indoor positioning is composed by the table of average measurements and the grid.

Next figures show the different results: Scanned Points attribute table and the Measurements Points Table.



OBJECTID	FLOOR	ID_ScanPoints	Shape *
12415	0	2	Point
12416	0	3	Point
12417	0	4	Point
12418	0	5	Point
12419	0	6	Point
12420	0	7	Point
12421	0	8	Point
12422	0	9	Point
12423	0	10	Point
12435	0	14	Point
12436	0	15	Point
12437	0	16	Point
12438	0	17	Point
12439	0	18	Point
12440	0	19	Point
12441	0	20	Point
12442	0	21	Point
12443	0	22	Point
12444	0	23	Point
13213	0	24	Point
13214	0	25	Point
13215	0	26	Point
13216	0	27	Point
13217	0	28	Point
13218	0	29	Point
13219	0	30	Point
13220	0	31	Point
13221	0	32	Point
13222	0	33	Point
13223	0	34	Point
13224	0	35	Point
13225	0	36	Point

Figure 42: Attribute table of the Wi-Fi Scan Points

OBJECTID: 9212
ID_POINT: 2
MAC: 00:13:c3:44:d8:21
INTENSITY: -71

OBJECTID: 9213
ID_POINT: 2
MAC: 00:14:69:f2:31:41
INTENSITY: -82

OBJECTID: 9214
ID_POINT: 2
MAC: 00:14:1c:dd:fc:41
INTENSITY: -92

OBJECTID: 9215
ID_POINT: 2
MAC: 00:25:86:d9:7b:76
INTENSITY: -73

OBJECTID: 9216
ID_POINT: 2
MAC: 00:14:69:f2:39:c0
INTENSITY: -55

OBJECTID: 9217
ID_POINT: 2
MAC: 00:13:c3:44:d8:20
INTENSITY: -70

OBJECTID: 9218
ID_POINT: 2
MAC: 00:1a:6d:9b:9a:20
INTENSITY: -84

OBJECTID: 9219
ID_POINT: 2
MAC: 00:14:69:f2:31:40
INTENSITY: -85

OBJECTID: 9220
ID_POINT: 2
MAC: 00:14:1c:dd:fc:40
INTENSITY: -92

OBJECTID: 9221
ID_POINT: 3
MAC: 00:14:69:f2:39:c1
INTENSITY: -56

OBJECTID: 9222
ID_POINT: 3
MAC: 00:1a:6d:9b:9a:21
INTENSITY: -91

OBJECTID: 9223
ID_POINT: 3
MAC: 00:14:69:f2:31:41
INTENSITY: -82

OBJECTID: 9224
ID_POINT: 3
MAC: 00:14:1c:dd:fc:41
INTENSITY: -92

OBJECTID: 9225
ID_POINT: 3
MAC: 00:25:86:d9:7b:76
INTENSITY: -73

OBJECTID: 9226
ID_POINT: 3
MAC: 00:13:c3:44:d8:20
INTENSITY: -70

OBJECTID: 9227
ID_POINT: 3
MAC: 00:1a:6d:9b:9a:20
INTENSITY: -84

OBJECTID: 9228
ID_POINT: 3
MAC: 00:14:69:f2:31:40
INTENSITY: -85

OBJECTID: 9229
ID_POINT: 3
MAC: 00:14:1c:dd:fc:40
INTENSITY: -92

OBJECTID: 9230
ID_POINT: 4
MAC: 00:13:c3:44:d8:21
INTENSITY: -71

OBJECTID: 9231
ID_POINT: 4
MAC: 00:1a:6d:9b:9a:21
INTENSITY: -91

OBJECTID: 9232
ID_POINT: 4
MAC: 00:14:69:f2:39:c1
INTENSITY: -56

OBJECTID: 9233
ID_POINT: 4
MAC: 00:14:69:f2:31:41
INTENSITY: -82

OBJECTID: 9234
ID_POINT: 4
MAC: 00:14:1c:dd:fc:41
INTENSITY: -92

OBJECTID: 9235
ID_POINT: 4
MAC: 00:25:86:d9:7b:76
INTENSITY: -73

OBJECTID: 9236
ID_POINT: 4
MAC: 00:14:69:f2:39:c0
INTENSITY: -55

OBJECTID: 9237
ID_POINT: 4
MAC: 00:13:c3:44:d8:20
INTENSITY: -70

OBJECTID: 9238
ID_POINT: 4
MAC: 00:1a:6d:9b:9a:20
INTENSITY: -84

OBJECTID: 9239
ID_POINT: 4
MAC: 00:14:69:f2:31:40
INTENSITY: -85

Figure 43: Samples of measurements displayed by the REST api

About a priori objectives have done most of them. The main objective achieved is that the application provide a working and tested tool to carry out the mapping phase of the Wi-Fi Indoor Positioning System. Another objective accomplished is that the application provides an intuitive User Interface to navigate and visualize the UJI's cartography and maps (Indoor and Outdoor). By other hand this application is developed in a compatible way with the others Smart Campus Android applications.

PART III: CONCLUSION

CONTENTS

14-. CONCLUSIONS

14.1.WIFI INDOOR POSITIONING

The Wi-Fi Indoor Positioning System is a positioning system able to develop in the context of the UJI campus because there are an existent infrastructure of Wi-Fi Access Points. This system is able to provide the position within the buildings. The smart phones can use this system because these have a Wi-Fi receiver.

The application is developed for Android devices. There are many devices that uses this platform. These devices can be smart phones and tablets, but the most common are the smart phones.

The indoor positioning is a powerful and a basic tool to a Smart Campus. This tool to will be the base of many applications. Many campus services will be improved from this tool.

Working with the UJI network sometimes it has been a problem. The problematic has been that the network fells and the Android device needs a VPN to connect with the UJI net. Sometimes this VPN didn't work correctly and the connection have been interrupted.

14.2. ARCGIS SDK ANDROID

The ArcGIS SDK to android provide to the mobile phone applications same functionalities and libraries. This SDK allow develop applications with complex location services. There are many advantages to use this SDK:

- Creating powerful, lightweight GIS applications
- Fast to display
- Works with the entire ArcGIS system
- Users can access local data and web services
- Allows collect and report data
- Perform GIS analysis

14.3. SMART CAMPUS

As discussed above, the smart campus technologies offers same new tools to manage and to help to the campus people. To bring this services to as many users as possible, focus the applications to a smart phone users is the solution. Until now, there was same tools developed for the viscaUJI project addressed to web services or web applications. The viscaUJI mobile applications is one step beyond.

PART IV: FUTURE DIRECTIONS

CONTENTS

15-.FUTURE DIRECTIONS

The direct future direction of this work is develop the Positioning Subsystem. This subsystem will perform the second phase of the Wi-Fi Indoor Positioning System. In this phase the obtained signal at user's location are matched back to the fingerprint database. The user's location corresponds to the best match using a location estimation algorithms. Different algorithms and methods will be tested to find a methodology that provide to the system sufficient precision.

Actually the system is provides the intensity average of the WAPs inside a 3 x 3 grid. It has been proved that to work with average values give more accurate results (Martin, Vinyals, Friedland, & Bajcsy, 2010) but this system will be tested working with this type of grid, but other methodology will be needed depending of the results.

Other future direction it will be implement the server part in an open network. Doing this, the VPN connection in the mobile device would not be necessary.

The application will be enhanced in some ways. The first think that it is possible to improve is the perform automatically the averages of the intensities in each cell of the grid. Other think to improve is the automatic control of the application errors. In some cases the application sends a scan point and a fingerprint to the server and this point is not saved by the server or the application don't "know" if the point is saved. Then other point is measured and this point is registered by the same identifier. This error actually is solved in post process, but it will be better if this don't succeed that.

When these improvements will be developed, and the application will be tested with a great accuracy, the other step of this project is measure all buildings of the campus, and then integrate the positioning subsystem as a module in other mobile applications of the viscaUJI project.

PART V: BIBLIOGRAPHY

16-.BIBLIOGRAPHY

Android developer. (n.d.). Retrieved October 2012, from http://developer.android.com/about/dashboards/index.html

Báez, M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., et al. *Introducción a Android.*

Caragliu, A., Del Bo, C., & Nijkamp, P. (2009). Smart cities in Europe.

Coll, E., Martínez, J., & Irigoyen, J. (2005). Los Sistemas de Información Geográfica en el ámbito municipal en la Provincia de Valencia. *Revista Iberoamericana de Automática e Informática Industrial* .

Coll, E., Martínez, J., & Irigoyen, J. (2004). Soluciones potenciales de SIG libres para cubrir las necesidades de gestión municipal en ayuntamientos medianos y pequeños.

Coll, E., Martínez, J., Ibarz, M., & Elgezaba, A. (2007). IDE local: Estudio de las necesidades de los ayuntamientos.

ESRI. (n.d.). *Resources ArcGIS*. Retrieved January 2013, from http://resources.arcgis.com/en/communities/runtime-android/

ESRI. (n.d.). *Resources ESRI REST API*. Retrieved January 2013, from http://resources.esri.com/help/9.3/arcgisserver/apis/rest/

Fundación Telefónica. (2011). *Smart Cities: un primer paso hacia la internet de las cosas.*

Geotec. (n.d.). Retrieved January 2013, from http://www.geotec.uji.es

Halder, S. J., & Kim, W. (2012). A Fusion Approach of RSSI and LQI for Indoor Localization. *Journal of Computer Networks and Communications* .

Hernández Muñoz, J. (2011). ¿Qué son las Smart Cities o Ciudades Inteligentes?

Kee, C., Yun, D., Jun, H., Parkinson, B., Pullen, S., & Lagestein, T. (2001). Centimeter-Accuracy Indoor Navigation Using GPS-Like Pseudolites . *GPS World* .

Le, M. H. (2009). *Indoor Navigation System for Handheld Devices.*

Lenihan, N. (2004). A local Optimal User Positioning System for Indoor Wireless Devices.

Li, B., Salter, J., Dempster, A., & Rizos, C. (2006). Indoor Positioning Techniques Based on Wireless LAN.

Liu, L., Zhang, J., Yi, Y., Li, H., & Zhang, J. (2012). Combating Interference: MU-MIMO, CoMP, and HetNet. *MITSUBISHI ELECTRIC RESEARCH LABORATORIES* .

Luna, A. G. (2012, 08). *Teknautas*. Retrieved January 2013, from http://www.elconfidencial.com/tecnologia/2012/08/23/android-contra-ios-que-sistema-operativo-es-mejor-para-usted-3010/

Martin, E., Vinyals, O., Friedland, G., & Bajcsy, R. (2010). Precise Indoor Localization Using Smart Phones.

Martínez, J., Coll, E., & Irigoyen, J. (2004). *Aplicación piloto de las necesidades municipales mediante Sistemas de Integración Geográfica.*

Meng, W., Xiao, W., Ni, W., & Xie, L. (2011). Secure and Robust WiFi Fingerprinting Indoor Localization.

Monoj-Kumar-Raja. (2010). *Monoj-Kumar-Raja*. Retrieved Enero 2013, from Wi-Fi Indoor Postioning System (Wi-Fi IPS): https://sites.google.com/site/monojkumarraja/academic-projects/wi-fi-indoor-positioning-system

Oredope, A., Gluhak, A., & Evans, B. (n.d.). *University of Surrey*. Retrieved January 2013, from http://www.surrey.ac.uk/ccsr/activity/facilities/futureinternet/index.htm

Pareja, E. (2012, September). *Ingeniería para optimistas*. Retrieved January 2013, from http://ingenieriaparaoptimistas.blogspot.com.es/2012/09/smart-cities-i-la-era-de-las-ciudades.html

Peñarrubia, J., & Rubio, F. (2007). Plan Estratégico de Evolición SIG en la Administración Local de la provincia de Valencia.

Poder PDA. (n.d.). Retrieved January 2013, from http://www.poderpda.com/plataformas/

Sanchis, A., Arnal, A., Moreno, W., Sanchis, V., Diaz, L., Huerta, J., et al. (2012). *viscaUJI: campus inteligente como IDE local.*

Schaffers, H., Komninos, N., Pallot, M., Trousse, B., Nilsson, M., & Oliveira, A. (2011). Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation. 431-446.

Serna, D. (2011). *Daniela Serna Blogspot*. Retrieved January 2013, from http://danielabloger-daniela.blogspot.com.es/2011/05/el-gps.html

Shapiro, J. (2005). Smart Cities: quality of life, productivity and the growth effects of human capital. *National Bureau of Economic Research* .

Thurston, J. (2002). GALILEO, GLONASS and NAVSTAR a report on GPS for GIS People.

Wang, Y., Jia, X., Lee, H., & Li, G. Y. (2003). An indoor wireless positioning system based on wireless local area network infrastructure.

Wikipedia. (n.d.). Retrieved January 2013, from http://ca.wikipedia.org/wiki/Android

Woo, S., Jeong, S., Mok, E., Xia, L., Choi, C., Pyeon, M., et al. (2011). Application of Wi-Fi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR. *Automation in construction 20* , 3-13.

Zeimpekis, V., Giaglis, G. M., & Lekakos, G. (2003). A Taxonomy of Indoor and Outdoor Positioning Techniques for Mobile Location Services.

PART VI: ANEXES

CONTENTS

17-. ANEXE I: SOURCE CODE

17.1. Clases

INDOORPOSITIONING.JAVA

```
package com.viscaUji.Indoor;

import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.util.ArrayList;
import java.util.List;
import java.util.Map;

import org.apache.http.impl.conn.tsccm.WaitingThread;
import org.codehaus.jackson.JsonNode;

import android.R.string;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.ColorFilter;
import android.graphics.Picture;
import android.graphics.drawable.Drawable;
import android.graphics.drawable.DrawableContainer;
import android.hardware.Camera.OnZoomChangeListener;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.wifi.ScanResult;
import android.net.wifi.WifiManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.util.TypedValue;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;


import com.esri.android.map.GraphicsLayer;
import com.esri.android.map.Layer;
import com.esri.android.map.MapView;
```

```
import com.esri.android.map.ags.ArcGISDynamicMapServiceLayer;
import com.esri.android.map.ags.ArcGISFeatureLayer;
import com.esri.android.map.ags.ArcGISImageServiceLayer;
import com.esri.android.map.ags.ArcGISLayerInfo;
import com.esri.android.map.ags.ArcGISFeatureLayer.MODE;
import com.esri.android.map.ags.ArcGISFeatureLayer.Options;
import com.esri.android.map.ags.ArcGISFeatureLayer.SELECTION_METHOD;
import com.esri.android.map.ags.ArcGISTiledMapServiceLayer;
import com.esri.android.map.event.OnZoomListener;
import com.esri.android.map.MapOnTouchListener;


import com.esri.core.geometry.Envelope;
import com.esri.core.geometry.Geometry;
import com.esri.core.geometry.GeometryEngine;
import com.esri.core.geometry.MultiPath;
import com.esri.core.geometry.Point;
import com.esri.core.geometry.Polygon;
import com.esri.core.geometry.Polyline;
import com.esri.core.map.CallbackListener;
import com.esri.core.map.FeatureEditResult;
import com.esri.core.map.FeatureSet;
import com.esri.core.map.FeatureTemplate;
import com.esri.core.map.FeatureType;
import com.esri.core.map.Field;
import com.esri.core.map.Graphic;
import com.esri.core.symbol.PictureMarkerSymbol;
import com.esri.core.symbol.SimpleFillSymbol;
import com.esri.core.symbol.SimpleLineSymbol;
import com.esri.core.symbol.SimpleMarkerSymbol;
import com.esri.core.symbol.SimpleMarkerSymbol.STYLE;
import com.esri.core.symbol.Symbol;
import com.esri.core.tasks.ags.query.Query;
import com.esri.core.tasks.ags.query.QueryTask;
import com.viscaUji.Indoor.R.drawable;

public class indoorpositioning extends Activity{

	WifiManager myWifiManager;
	boolean wasEnabled;
	List<ScanResult> wireless;
	ScanResult WPA;
	ArrayList<pto_escaneado> puntosEscaneados = new ArrayList<pto_escaneado>();
	public Location loc;
	int numPto = 0;

	ToggleButton tbAddAntenna, tbAddScan;
	boolean addAntennaChecked, addScanChecked;
	Button bSave;

	MyTouchListener myListener = null;

	MapView mMapView ;
	Layer baseMap, viscaUjiColorLayer, viscaUjiDesatLayer, grid3x3;
	ArcGISFeatureLayer ujiAntenna, ujiScans, ujiMeasurements;
	ArcGISDynamicMapServiceLayer ujiBuildInt, ujiBuildings;
	GraphicsLayer gAntenna, gScan;

	//id Routers and Scaner Measurements and points
```

```
        int idAntenna, idScan = 10, idMeasurement;
        //Attributes

        int[] layerFloorsId = {0,3,6,9,12,15,18,21};
        //Routers
        antennaDataFile AntennaDatFil;

        ArrayList<Point> scanPoints = new ArrayList<Point>();
        ArrayList<Point> points = new ArrayList<Point>();
        ArrayList<myPoint> rterPoints = new ArrayList<myPoint>();
        //Añadir Routers
        ArrayList<antennaWifi> antennas = new ArrayList<antennaWifi>();

        int floor;
        String build;
        double scale;
        TextView tvFloor, tvInfo;
        String textFloor = "   Floor nº: ";

        private CallbackListener<FeatureSet> callback = new
CallbackListener<FeatureSet>() {

                public void onCallback(FeatureSet objs) {
                        // TODO Auto-generated method stub
                }

                public void onError(Throwable e) {
                        // TODO Auto-generated method stub

                }
        };

        String urlAntenna =
"http://mastergeotech.dlsi.uji.es/arcgis/rest/services/WifiIndoor/Antenna/FeatureServer
/0";
        String urlScans =
"http://mastergeotech.dlsi.uji.es/arcgis/rest/services/WifiIndoor/WiFi_ScanPoints/Featu
reServer/0";
        String urlMeasurements =
"http://mastergeotech.dlsi.uji.es/arcgis/rest/services/WifiIndoor/WiFi_ScanPoints/Featu
reServer/1";
        String urlGrid =
"http://mastergeotech.dlsi.uji.es/arcgis/rest/services/WifiIndoor/Grids_Editing/MapServ
er";

        protected String editingerrormessage;

     /** Called when the activity is first created. */
     @Override
     public void onCreate(Bundle savedInstanceState) {
         super.onCreate(savedInstanceState);
         setContentView(R.layout.indoorpositioning);

         tvFloor = (TextView)findViewById(R.id.tvFloor);
         tvInfo = (TextView)findViewById(R.id.tvInfo);

         tbAddAntenna = (ToggleButton)findViewById(R.id.tbAddAntenna);
         tbAddScan = (ToggleButton)findViewById(R.id.tbAddScan);
```

```
        mMapView = (MapView)findViewById(R.id.map);
        mMapView.setExtent(new Envelope(-9001.029094706248, 4864711.126100904,-
6827.351102789284, 4866019.55363002), 0);
             mMapView.setEsriLogoVisible(true);

             myListener = new MyTouchListener(indoorpositioning.this, mMapView);
             mMapView.setOnTouchListener(myListener);

             //Encender el wifi
             myWifiManager = (WifiManager) getSystemService(Context.WIFI_SERVICE);
           wasEnabled = myWifiManager.isWifiEnabled();

           if (!wasEnabled){
                 myWifiManager.setWifiEnabled(true);
                 Toast.makeText(this, "Wi-Fi is disabled", Toast.LENGTH_SHORT).show();
           }else {
              Toast.makeText(this, "Wi-Fi is enabled", Toast.LENGTH_SHORT).show();
           }

             viscaUjiColorLayer = new ArcGISTiledMapServiceLayer(
             "http://geotec.dlsi.uji.es/arcgis/rest/services/UJI/viscaUJI/MapServer");
             mMapView.addLayer(viscaUjiColorLayer);

             viscaUjiDesatLayer = new ArcGISTiledMapServiceLayer(

       "http://geotec.dlsi.uji.es/arcgis/rest/services/UJI/viscaUJI_desaturado/MapServer
");
             viscaUjiDesatLayer.setVisible(false);
             mMapView.addLayer(viscaUjiDesatLayer);

             ujiBuildings = new
ArcGISDynamicMapServiceLayer("http://geotec.dlsi.uji.es/arcgis/rest/services/UJI/Buildi
ngs/MapServer");
        ujiBuildings.setVisible(true);

             ujiBuildInt = new
ArcGISDynamicMapServiceLayer("http://mastergeotech.dlsi.uji.es/arcgis/rest/services/Wif
iIndoor/BuildingInteriorbyFloor/MapServer");
             ujiBuildInt.setVisible(false);

             grid3x3 = new ArcGISDynamicMapServiceLayer(urlGrid);
             grid3x3.setVisible(false);

             ujiAntenna = new ArcGISFeatureLayer(urlAntenna, MODE.SELECTION);
             ujiScans = new ArcGISFeatureLayer(urlScans, MODE.ONDEMAND);
             ujiAntenna.setVisible(false);
             ujiScans.setVisible(false);
             ujiMeasurements = new ArcGISFeatureLayer(urlMeasurements, MODE.ONDEMAND);

        mMapView.addLayer(ujiBuildings);
        mMapView.addLayer(ujiBuildInt);
        mMapView.addLayer(grid3x3);

        gAntenna = new GraphicsLayer();
        gScan = new GraphicsLayer();

        mMapView.addLayer(gAntenna);
        mMapView.addLayer(gScan);
        mMapView.addLayer(ujiAntenna);
```

```
        mMapView.addLayer(ujiScans);

        //Antenna data
        AntennaDatFil = new antennaDataFile();
             int[] antennasFloor = {0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,0,0,1,1,1,2,2,3,3,3};
             AntennaDatFil.setFloor(antennasFloor);
             String[] antennasNames = {"pa-ti19","pa-ti17","pa-ti18","pa-ti16","pa-
ti113","pa-ti14","pa-ti15","pa-ti13","pa-ti11","pa-ti12","pa-ti110","pa-ti111","pa-
ti112","pa-ti114","pa-ti25","pa-ti26","pa-ti23","pa-ti24","pa-ti29","pa-ti21","pa-
ti22","pa-ti210","pa-ti27","pa-ti28"};
             AntennaDatFil.setRoutersName(antennasNames);

        tbAddAntenna.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
                   if (isChecked) {
                        // The toggle is enabled
                         addAntennaChecked = true;
                         ujiAntenna.setVisible(true);
                         tbAddScan.setClickable(false);
                         tvInfo.setText("Tap on the map to add a Wi-Fi Antenna");
                         Toast.makeText(getApplicationContext(), "Add Antenna",
Toast.LENGTH_SHORT).show();
                    } else {
                        // The toggle is disabled
                         tbAddScan.setClickable(true);
                         addAntennaChecked = false;
                         ujiAntenna.setVisible(false);
                         tvInfo.setText("...");
                    }
                }
        });

        tbAddScan.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked)
{
                if(ujiBuildInt.isVisible()){
                   if (isChecked) {
                        // The toggle is enabled
                         grid3x3.setVisible(true);
                                ujiScans.setVisible(true);
                         addScanChecked = true;
                         tbAddAntenna.setClickable(false);
                         tvInfo.setText("Tap on the map to add a Figerprint");
                         Toast.makeText(getApplicationContext(), "Add Scan",
Toast.LENGTH_SHORT).show();
                   } else {
                        // The toggle is disabled
                   grid3x3.setVisible(false);
                                ujiScans.setVisible(false);
                         tbAddAntenna.setClickable(true);
                         addScanChecked = false;
                         tvInfo.setText("...");
                    }
                }
            }
        });
```

```
    }

	private void readAttirbuteWifiPoints() {
		// TODO Auto-generated method stub
		int id[] = ujiScans.getGraphicIDs();
		int lenght = id.length;
//		Toast.makeText(indoorpositioning.this, "lenght  "+ lenght,
Toast.LENGTH_SHORT).show();
		int lastID=id[lenght-1];
		Graphic g = ujiScans.getGraphic(lastID);
		idScan = (Integer) g.getAttributeValue("ID_ScanPoints") + 1;
//		Toast.makeText(indoorpositioning.this, "last id "+ (idScan-1),
Toast.LENGTH_SHORT).show();
//		Toast.makeText(indoorpositioning.this, "new id "+ idScan,
Toast.LENGTH_SHORT).show();
	}

	private void readAttributeAntenna() {
		// TODO Auto-generated method stub
		int numberAntenna = ujiAntenna.getNumberOfGraphics();
		idAntenna = numberAntenna;
	}

	class MyTouchListener extends MapOnTouchListener {
		// ArrayList<Point> polylinePoints = new ArrayList<Point>();
		MultiPath poly;
		String type = "";
		Point startPoint = null;

		public MyTouchListener(Context context, MapView view) {
			super(context, view);
		}

		public void setType(String geometryType) {
			this.type = geometryType;
		}

		public String getType() {
			return this.type;
		}

		/*
		 * com.esri.android.map.MapOnTouchListener#onSingleTap(android.view.
		 * MotionEvent)
		 */
		public boolean onSingleTap(MotionEvent e) {
			if(tbAddAntenna.isChecked() || tbAddScan.isChecked()){
				if(ujiBuildInt.isVisible()){
					if(addAntennaChecked){
						readAttributeAntenna();
						addAntennaInformation(e);

					}else if(addScanChecked){
						addScanInformation(e);

					}else{
						Toast.makeText(indoorpositioning.this, "Choose
between add scan or add antenna", Toast.LENGTH_SHORT).show();
					}
```

```
                         }else {
                                Toast.makeText(getApplicationContext(), "You have to
initialize the Buildings Interiors Layer", Toast.LENGTH_SHORT).show();
                         }
                  }
                  return false;
            }
    }

      public void addScanInformation(MotionEvent e) {
            // TODO Auto-generated method stub
            myPoint point = new myPoint(e.getX(),e.getY(),floor);
            ArrayList<wpas> wifisAux = new ArrayList<wpas>();
            wifisAux = scanWifi();

            //Attributes Feature WifiScan
            String floor = "FLOOR", idScanPoint = "ID_ScanPoints";
            Graphic g = ujiScans.createFeatureWithTemplate(null,
mMapView.toMapPoint(new Point(e.getX(), e.getY())));
            readAttirbuteWifiPoints();
            Map<String,Object> attributesPointScan = g.getAttributes();
            attributesPointScan.put(floor, point.getFloor());
            attributesPointScan.put(idScanPoint, idScan);
            Graphic newGraphic = new Graphic(mMapView.toMapPoint(new Point(e.getX(),
e.getY())),g.getSymbol(),
            attributesPointScan, g.getInfoTemplate());
            Graphic[] adds = new Graphic[] {newGraphic};
            ujiScans.applyEdits(adds, null, null, null);
            ujiScans.refresh();

//          Attributes WifiMeasurement Table
            Graphic m = ujiMeasurements.createFeatureWithTemplate(null, null);
            String  idPoint = "id_point", name = "Name", mac = "Mac", intens =
"Intensity";
            Map<String,Object> attributesMeasurements = m.getAttributes();
            int cont = wifisAux.size();
            wpas aux;
            Graphic newGraphic2;

            for (int i=0; i<cont; i++){
                  aux = wifisAux.get(i);
                  attributesMeasurements.put(idPoint, idScan);
                  attributesMeasurements.put(name, aux.getNombre());
                  attributesMeasurements.put(mac, aux.getMac());
                  attributesMeasurements.put(intens, aux.getIntensidad());

                  newGraphic2 = new Graphic(null, null, attributesMeasurements,
m.getInfoTemplate());
                  Graphic[] adds2 = new Graphic[] {newGraphic2};
                  ujiMeasurements.applyEdits(adds2, null, null, null);
            }

      }

      public void addAntennaInformation(MotionEvent e) {
            // TODO Auto-generated method stu
            final MotionEvent ei = e;
            final antennaWifi antennaAux = new antennaWifi();
            AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(this);
```

```
            dialogBuilder.setTitle("Select the router");
            final String[] antennas = AntennaDatFil.getNamesFile();
            final int[] floorAntennas = AntennaDatFil.getFloors();
            dialogBuilder.setItems(antennas, new DialogInterface.OnClickListener() {

                  public void onClick(DialogInterface dialog, int which) {
                        // TODO Auto-generated method stub
                        AntennaDatFil.setNameAntenna(antennas[which]);
                        AntennaDatFil.setCont(which);
                        AntennaDatFil.setFloorAntenna(floorAntennas[which]);

                        boolean bfloor = AntennaDatFil.checkFloorMapFile(floor);
                        if(bfloor){
                              myPoint auxPR = new myPoint(ei.getX(),ei.getY(),floor);
                              antennaAux.setPto(auxPR);
                              antennaAux.setId(idAntenna);
                              antennaAux.setNombre(AntennaDatFil.getNameAntenna());
                              antennaAux.setPto(auxPR);
                              Graphic g = ujiAntenna.createFeatureWithTemplate(null,
mMapView.toMapPoint(new Point(ei.getX(), ei.getY())));

                              String name = "Name", floor = "Floor";
                              Map<String,Object> attributes = g.getAttributes();
                              attributes.put(name, antennaAux.getNombre());
                              attributes.put(floor, antennaAux.getPto().getFloor());

                              Graphic graphic = new Graphic(mMapView.toMapPoint(new
Point(ei.getX(), ei.getY())), new SimpleMarkerSymbol(Color.RED,10,STYLE.CIRCLE));
                              Graphic newGraphic = new
Graphic(mMapView.toMapPoint(new Point(ei.getX(), ei.getY())),g.getSymbol(),attributes,
g.getInfoTemplate());

                              gAntenna.addGraphic(graphic);
                              Graphic[] adds = new Graphic[] {newGraphic};

                              ujiAntenna.applyEdits(adds, null, null, null);
                              ujiAntenna.refresh();
                              gAntenna.removeAll();
                        }else{
                              Toast.makeText(indoorpositioning.this, "This antenna is
in the floor:" +
AntennaDatFil.getAntennaFloorByPosition(which),Toast.LENGTH_SHORT).show();
                        }
                  }
            });

                  AlertDialog alertDialog = dialogBuilder.create();
                  alertDialog.show();
      }

      public boolean onCreateOptionsMenu(Menu menu){
            super.onCreateOptionsMenu(menu);
            MenuInflater awesome = getMenuInflater();
            awesome.inflate(R.menu.main_menu, menu);
            return true;
      }

    public ArrayList<wpas> scanWifi(){
```

```
            Toast.makeText(indoorpositioning.this, "Scanning WiFi nets...",
Toast.LENGTH_SHORT).show();
            wireless = myWifiManager.getScanResults(); // Returns a <list> of
scanResults
            int num_WPA = wireless.size();
          String nombre_WPA, mac;
          int intensity;
          wpas aux1;
          ArrayList<wpas> wifisAux = new ArrayList<wpas>();
          for (int i = 0; i<num_WPA; i++){
            WPA= wireless.get(i);
            nombre_WPA = WPA.SSID;
            intensity = WPA.level;
            mac = WPA.BSSID;
            aux1 = new wpas(idMeasurement, idScan, nombre_WPA, mac, intensity);
            wifisAux.add(i, aux1);
            idMeasurement++;
          }
          return wifisAux;
      }

    public boolean onOptionsItemSelected(MenuItem item){
      switch (item.getItemId()){
            case R.id.menuBaseMap:
                  Toast.makeText(this, "Seleccionar BaseMap",
Toast.LENGTH_SHORT).show();
                  final CharSequence[] items = {"COLOR BaseMap", "GREY BaseMap"};
                  AlertDialog.Builder builder = new AlertDialog.Builder(this);
                  builder.setTitle("How do you want visualize the BaseMap?");
                  builder.setSingleChoiceItems(items, -1, new
DialogInterface.OnClickListener() {
                      public void onClick(DialogInterface dialog, int item) {
                          Toast toast = Toast.makeText(getApplicationContext(), "Haz
elegido la opcion: " + items[item] , Toast.LENGTH_SHORT);
                          toast.show();
                          dialog.cancel();
                          Boolean visibleColor = viscaUjiColorLayer.isVisible();
                          Boolean visibleGrey = viscaUjiDesatLayer.isVisible();
                          switch (item){
                          case 0:
                                if(!visibleColor){
                                      viscaUjiDesatLayer.setVisible(false);
                                      viscaUjiColorLayer.setVisible(true);
                                }
                                Toast.makeText(getApplicationContext(), "ColorBaseMap",
Toast.LENGTH_SHORT).show();
                                break;
                          case 1:
                                if(!visibleGrey){
                                      viscaUjiDesatLayer.setVisible(true);
                                      viscaUjiColorLayer.setVisible(false);
                                }
                                Toast.makeText(getApplicationContext(), "GreyBaseMap",
Toast.LENGTH_SHORT).show();
                                break;
                          }
                      }
                  });
                  AlertDialog alert = builder.create();
```

```
                    alert.show();
                    return true;
             case R.id.menuBuildInt:
                    //Añadir la capa de Buildings Interiors,
                    Toast.makeText(this, "Seleccionar BaseMap",
Toast.LENGTH_SHORT).show();
                    final CharSequence[] items2 = {"Buildings", "Buildings Interiors"};
                    AlertDialog.Builder builder2 = new AlertDialog.Builder(this);
                    builder2.setTitle("Select: Buildings or Buildings Interiors");
                    builder2.setSingleChoiceItems(items2, -1, new
DialogInterface.OnClickListener() {
                       public void onClick(DialogInterface dialog, int item) {
                            Toast toast = Toast.makeText(getApplicationContext(), "Haz
elegido la opcion: " + items2[item] , Toast.LENGTH_SHORT);
                            toast.show();
                            dialog.cancel();
                            Boolean buildVisible = ujiBuildings.isVisible();
                            Boolean buildIntVisible = ujiBuildInt.isVisible();

                            switch (item){
                            case 0:
                                 if(!buildVisible){
                                        ujiBuildInt.setVisible(false);
                                        ujiBuildings.setVisible(true);
                                 }
                                 break;
                            case 1:
                                 if(!buildIntVisible){
                                        ujiBuildings.setVisible(false);
                                        ujiBuildInt.setVisible(true);
                                        floor = 0;
                                        writeFloor();
                                        selectfloor();
                                 }
                                 break;
                            }
                        }
                    });
                    AlertDialog alert2 = builder2.create();
                    alert2.show();
                    return true;
             case R.id.menuFloor:
                    //Canviar de planta
                    if(ujiBuildInt.isVisible()){
                           final CharSequence[] items3 = {"S Floor","0 Floor", "1st
Floor", "2nd Floor","3th Floor", "4th Floor","5th Floor","6th Floor"};
                           AlertDialog.Builder builder3 = new AlertDialog.Builder(this);
                           builder3.setTitle("Select the floor of the Building
Interior:...");
                           final ArcGISLayerInfo[] dynamInfoLayers =
ujiBuildInt.getAllLayers();
                           builder3.setSingleChoiceItems(items3, -1, new
DialogInterface.OnClickListener() {
                               public void onClick(DialogInterface dialog, int item) {
                                   Toast toast = Toast.makeText(getApplicationContext(),
"Haz elegido la opcion: " + items3[item] , Toast.LENGTH_SHORT);
                                   toast.show();
                                   dialog.cancel();
                                   switch (item){
```

```
                        case 0:
                            floor = -1;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[0]].setVisible(true);
                            ujiBuildInt.refresh();
                            selectfloor();
                            writeFloor();
                            break;
                        case 1:
                            floor = 0;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[1]].setVisible(true);
                            ujiBuildInt.refresh();
                            selectfloor();
                            writeFloor();
                            break;
                        case 2:
                            floor = 1;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[2]].setVisible(true);
                            ujiBuildInt.refresh();
                            selectfloor();
                            writeFloor();
                            break;
                        case 3:
                            floor = 2;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[3]].setVisible(true);
                            ujiBuildInt.refresh();
                            selectfloor();
                            writeFloor();
                            break;
                        case 4:
                            floor = 3;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[4]].setVisible(true);
                            ujiBuildInt.refresh();
                            selectfloor();
                            writeFloor();
                            break;
                        case 5:
                            floor = 4;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[5]].setVisible(true);
                            ujiBuildInt.refresh();
                            selectfloor();
                            writeFloor();
                            break;
                        case 6:
                            floor = 5;
                            desactiveAllLayers();

    dynamInfoLayers[layerFloorsId[6]].setVisible(true);
```

```
                                    ujiBuildInt.refresh();
                                    selectfloor();
                                    writeFloor();
                                    break;
                               case 7:
                                    floor = 6;
                                    desactiveAllLayers();

       dynamInfoLayers[layerFloorsId[7]].setVisible(true);
                                    ujiBuildInt.refresh();
                                    selectfloor();
                                    writeFloor();
                                    break;
                               }
                           }
                       });
                       AlertDialog alert3 = builder3.create();
                       alert3.show();
                       Toast.makeText(this, "Seleccionar Floor",
Toast.LENGTH_SHORT).show();
                   } else{
                       Toast.makeText(this, "You have to activate the Building
Interior layer", Toast.LENGTH_SHORT).show();
                   }
                   return true;
             case R.id.menuAbout:

                   Intent indoor = new Intent(indoorpositioning.this, about.class);
                   startActivity(indoor);

//                 startActivity(new Intent("com.viscaUJI.Indoor.ABOUT"));
                   return true;
             }
             return false;
    }
    protected void selectfloor() {
             // TODO Auto-generated method stub
       ujiAntenna.clear();
             ujiAntenna.clearSelection();
             Query pAntennasScans = new Query();
             String where = "FLOOR = '"+ floor +"'";
             pAntennasScans.setWhere(where);
             pAntennasScans.setReturnGeometry(true);
             pAntennasScans.setInSpatialReference(mMapView.getSpatialReference());
             ujiAntenna.selectFeatures(pAntennasScans, SELECTION_METHOD.NEW, callback);
       }

       protected void desactiveAllLayers() {
             // TODO Auto-generated method stub
       int lenght = layerFloorsId.length;
       ArcGISLayerInfo[] dynamInfoLayers = ujiBuildInt.getAllLayers();
       for (int i = 0; i<lenght; i++){
                   dynamInfoLayers[layerFloorsId[i]].setVisible(false);
       }
       }

       protected void writeFloor(){
       String floorTxt = textFloor + floor;
             tvFloor.setText(floorTxt);
```

```
    }

        @Override
        protected void onDestroy() {
              super.onDestroy();
 }
        @Override
        protected void onPause() {
              super.onPause();
              mMapView.pause();
 }
        @Override    protected void onResume() {
              super.onResume();
              mMapView.unpause();
        }
}
```

MAIN.JAVA

```
package com.viscaUji.Indoor;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.Button;
import android.widget.ImageView;

public class main extends Activity implements OnTouchListener{

	ImageView imgStart;

	public void onCreate(Bundle savedInstanceState) {
		 super.onCreate(savedInstanceState);
		setContentView(R.layout.main);
		imgStart = (ImageView)findViewById(R.id.ivStart);
		 imgStart.setOnClickListener(new View.OnClickListener() {

			public void onClick(View v) {
				// TODO Auto-generated method stub
				switch (v.getId()){
				case (R.id.ivStart):
					Intent indoor = new Intent(main.this, indoorpositioning.class);
					startActivity(indoor);
					break;
				}
			}
		});
      imgStart.setOnClickListener(new View.OnClickListener() {

			public void onClick(View v) {
				// TODO Auto-generated method stub
				switch (v.getId()){
				case (R.id.ivStart):
					Intent indoor = new Intent(main.this, indoorpositioning.class);
					startActivity(indoor);
```

```
                        break;
                    }
                }
            });
        }

   @Override
   public boolean onCreateOptionsMenu(Menu menu) {
     //getMenuInflater().inflate(R.menu., menu);
     return true;
   }

        public boolean onTouch(View v, MotionEvent event) {
            // TODO Auto-generated method stub
            return false;
        }
        public boolean onKeyDown(int keyCode, KeyEvent event){

            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setMessage("Are you sure you want to exit?")
            .setCancelable(false)
            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    main.this.finish();
                }
            })
            .setNegativeButton("No", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }
            });
            AlertDialog alert = builder.create();
            alert.show();
        return super.onKeyDown(keyCode, event);
}
}
```

ANTENNADATAFILE.JAVA

```
package com.viscaUji.Indoor;

public class antennaDataFile {

	private String[] antennaNames;
	int[] floors;
	String nameAntenna;
	int floorAntenna;
	private int cont;

	public antennaDataFile() {
		super();
		// TODO Auto-generated constructor stub
	}
	public String[] getNamesFile() {
		return antennaNames;
	}
	public void setRoutersName(String[] antennaNames) {
		this.antennaNames = antennaNames;
	}
	public int[] getFloors() {
		return floors;
	}
	public void setFloor(int[] floor) {
		this.floors = floor;
	}
	public int getCont() {
		return cont;
	}
	public void setCont(int cont) {
		this.cont = cont;
	}
	public String getNameAntenna() {
		return nameAntenna;
	}
	public void setNameAntenna(String nameAntenna) {
		this.nameAntenna = nameAntenna;
	}
	public int getFloorAntenna() {
		return floorAntenna;
	}
	public void setFloorAntenna(int floorAntenna) {
		this.floorAntenna = floorAntenna;
```

```
	}
	public void setData(){

	}
	public boolean checkFloorMapFile(int floorMap) {
		// TODO Auto-generated method stub
		boolean checkFloor;
		if (floorMap == floorAntenna){
			checkFloor = true;
		} else{
			checkFloor = false;
		}
		return checkFloor;
	}
}
```

ANTENNAWIFI.JAVA

```
package com.viscaUji.Indoor;

import android.graphics.PointF;

public class antennaWifi extends PointF {

	int id;
	String nombre;
	myPoint pto;

	public antennaWifi(int id, String nombre, myPoint pto) {
		super();
		this.id = id;
		this.nombre = nombre;
		this.pto = pto;
	}
	public antennaWifi() {
		super();
		// TODO Auto-generated constructor stub
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public String getNombre() {
		return nombre;
	}
	public void setNombre(String nombre) {
		this.nombre = nombre;
	}
	public myPoint getPto() {
		return pto;
	}
	public void setPto(myPoint pto) {
		this.pto = pto;
	}
}
```

ATTRIBUTEITEM.JAVA

```
package com.viscaUji.Indoor;

import android.view.View;
import com.esri.core.map.Field;

/**
 * POJO for storing the data associated with a row in the attributes list
 */
public class AttributeItem {

  private Field field;
  private Object value;
  private View view;

  public View getView() {
    return view;
  }

  public void setView(View view) {
    this.view = view;
  }

  public Field getField() {
    return field;
  }

  public void setField(Field field) {
    this.field = field;
  }

  public Object getValue() {
    return value;
  }

  public void setValue(Object value) {
    this.value = value;
  }
}
```

COORD3D.JAVA

```
package com.viscaUji.Indoor;

public class coord3D {
	double lat, lon, alt;

	public coord3D() {
		super();
		// TODO Auto-generated constructor stub
	}
	public coord3D(double lat, double lon, double alt) {
		super();
		this.lat = lat;
		this.lon = lon;
		this.alt = alt;
	}
	public double getLat() {
		return lat;
	}

	public void setLat(double lat) {
		this.lat = lat;
	}

	public double getLon() {
		return lon;
	}

	public void setLon(double lon) {
		this.lon = lon;
	}

	public double getAlt() {
		return alt;
	}

	public void setAlt(double alt) {
		this.alt = alt;
	}
}
```

FEATURELAYERUTILS.JAVA

```
package com.viscaUji.Indoor;

public class coord3D {
	double lat, lon, alt;

	public coord3D() {
		super();
		// TODO Auto-generated constructor stub
	}

	public coord3D(double lat, double lon, double alt) {
		super();
		this.lat = lat;
		this.lon = lon;
		this.alt = alt;
	}

	public double getLat() {
		return lat;
	}

	public void setLat(double lat) {
		this.lat = lat;
	}

	public double getLon() {
		return lon;
	}

	public void setLon(double lon) {
		this.lon = lon;
	}

	public double getAlt() {
		return alt;
	}

	public void setAlt(double alt) {
		this.alt = alt;
	}
}
```

MYPOINT.JAVA

```
package com.viscaUji.Indoor;

import android.graphics.Point;

public class myPoint extends Point{

	double x, y;
	int floor;
	public myPoint() {
		super();
		// TODO Auto-generated constructor stub
	}
	public myPoint(double x, double y, int floor) {
		super();
		this.x = x;
		this.y = y;
		this.floor = floor;

	}
	public double getX() {
		return x;
	}
	public void setX(double x) {
		this.x = x;
	}
	public double getY() {
		return y;
	}
	public void setY(double y) {
		this.y = y;
	}
	public int getFloor() {
		return floor;
	}
	public void setFloor(int floor) {
		this.floor = floor;
	}

}
```

PTO_ESCANEADO.JAVA

```
package com.viscaUji.Indoor;

import android.graphics.Point;

public class myPoint extends Point{

	double x, y;
	int floor;
	public myPoint() {
		super();
		// TODO Auto-generated constructor stub
	}
	public myPoint(double x, double y, int floor) {
		super();
		this.x = x;
		this.y = y;
		this.floor = floor;

	}
	public double getX() {
		return x;
	}
	public void setX(double x) {
		this.x = x;
	}
	public double getY() {
		return y;
	}
	public void setY(double y) {
		this.y = y;
	}
	public int getFloor() {
		return floor;
	}
	public void setFloor(int floor) {
		this.floor = floor;
	}

}
```

WPAS.JAVA

```
package com.viscaUji.Indoor;

public class wpas {
	String nombre, mac;
	int intensidad, idPoint, idMeasure;

	public wpas() {
		super();
		// TODO Auto-generated constructor stub
	}

	public wpas(int idMeas, int idPt, String nombre, String mac, int intensidad) {
		super();
		this.idMeasure = idMeas;
		this.idPoint = idPt;
		this.nombre = nombre;
		this.mac = mac;
		this.intensidad = intensidad;
	}

	public int getIdMeasurement() {
		return idMeasure;
	}

	public void setIdMeasurement(int idMeas) {
		this.idPoint = idMeas;
	}

	public int getIdPoint() {
		return idPoint;
	}

	public void setIdPoint(int idPt) {
		this.idPoint = idPt;
	}

	public String getNombre() {
		return nombre;
	}

	public void setNombre(String nombre) {
		this.nombre = nombre;
	}
```

```
	public String getMac() {
		return mac;
	}

	public void setMac(String mac) {
		this.mac = mac;
	}

	public int getIntensidad() {
		return intensidad;
	}

	public void setIntensidad(int intensidad) {
		this.intensidad = intensidad;
	}
}
```

17.2. Layout

MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >

        <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical" >

                <ImageView
                        android:id="@+id/imageView1"
                        android:layout_width="fill_parent"
                        android:layout_height="fill_parent"
                android:src="@drawable/smartuji_banner" />

        </LinearLayout>
        <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_weight="0.93"
                android:orientation="vertical" >

                <TextView
                        android:id="@+id/textView1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_gravity="center_vertical|center_horizontal"
                        android:gravity="center"
                        android:text="Tap on the image to enter into the application"
                        android:textAppearance="?android:attr/textAppearanceLarge" />

                <ImageView
                        android:id="@+id/ivStart"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:src="@drawable/smart_campus" />

        </LinearLayout>
```

```
        <LinearLayout
                android:layout_width="318dp"
                android:layout_height="79dp"
                android:orientation="vertical" >

                <ImageView
                        android:id="@+id/imageView4"
                        android:layout_width="fill_parent"
                        android:layout_height="fill_parent"
                        android:src="@drawable/smatuji_thumb" />
        </LinearLayout>
</LinearLayout>
```

INDOORPOSITIONING.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    tools:context=".ViscaUjiIndoorPositioningActivity" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/bt_pos"
            style="?android:attr/buttonStyleSmall"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Position" />

        <TextView
            android:id="@+id/tvFloor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="  Floor nº:..."
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <ImageView
            android:id="@+id/imageView1"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:layout_gravity="right"
            android:src="@drawable/smatuji_b" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.68"
```

```xml
        android:orientation="vertical" >

        <com.esri.android.map.MapView
            android:id="@+id/map"
            android:layout_width="match_parent"
            android:layout_height="276dp"
            android:layout_weight="0.29" >
        </com.esri.android.map.MapView>

    </LinearLayout>

     <LinearLayout
         android:layout_width="match_parent"
         android:layout_height="wrap_content" >

         <ToggleButton
             android:id="@+id/tbAddAntenna"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"
             android:layout_weight="0.39"
             android:text="Add Router"
             android:textOff="@string/addRouter"
             android:textOn="@string/addRouter" />

         <ToggleButton
             android:id="@+id/tbAddScan"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"
             android:layout_weight="0.62"
             android:text="Scan Wifi"
             android:textOff="@string/wifiScan"
             android:textOn="@string/wifiScan" />

     </LinearLayout>

     <LinearLayout
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:orientation="vertical" >

         <TextView
             android:id="@+id/tvInfo"
             android:layout_width="wrap_content"
             android:layout_height="wrap_content"
             android:layout_gravity="center"
             android:text="..."
             android:textAppearance="?android:attr/textAppearanceMedium" />



     </LinearLayout>

</LinearLayout>
```

17.3. Menus, PopUps and Android Manifest

MAIN_MENU.XML

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
        <item
                android:id="@+id/menuBaseMap"
                android:alphabeticShortcut="b"
                android:title="Select BaseMap"
        />

        <item
                android:id="@+id/menuBuildInt"
                android:alphabeticShortcut="t"
                android:title="Add BuildInteriors Layer"
        />

        <item
                android:id="@+id/menuFloor"
                android:alphabeticShortcut="f"
                android:title="Select floor"
        />

         <item
                android:id="@+id/menuRouter"
                android:alphabeticShortcut="t"
                android:title="Add Routers"
        />

</menu>
```

ABOUT_MENU.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This applicattion have been developed by Joan Pere Avariento Vicent. The aim of the app is to provide a powerful tool to perform the indoor positioning in the UJI. This tool will be used for the viscaUJI project." />

</LinearLayout>
```

POPUPBASEMAP.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
            android:id="@+id/rgBaseMap"
            android:layout_width="match_parent"
            android:layout_height="match_parent" >
    <RadioButton
            android:id="@+id/rbBaseMapColor"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Color Base Map" />
    <RadioButton
            android:id="@+id/rbBaseMapGrey"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Grey Base Map" />
</RadioGroup>
```

POPUPBUILDING.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/rgBuildings"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <RadioButton
                android:id="@+id/rbBuild"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Buildings" />

        <RadioButton
                android:id="@+id/rbBuildInt"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Buildings Interior" />

</RadioGroup>
```

POPUPBUILDINT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >

        <CheckBox
                android:id="@+id/cb_IntSpaces"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Interior Spaces" />

        <CheckBox
                android:id="@+id/cb_FloorPlant"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Floor Plant" />

        <Button
                android:id="@+id/bt_load"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Load" />

</LinearLayout>
```

ANDROIDMANIFEST.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
	package="com.viscaUji.Indoor"
	android:versionCode="1"
	android:versionName="1.0">
	<uses-sdk android:minSdkVersion="8" />
	<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
	<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
	<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
	<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION" />
	<uses-permission android:name="android.permission.CONTROL_LOCATION_UPDATES" />
	<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
	<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>

	<uses-feature android:glEsVersion="0x00020000" android:required="true"/>

		<application android:icon="@drawable/ic_launcher" android:label="@string/app_name">
			<activity android:name=".main"
				android:label="Smart Uji Indoor Positioning">
				<intent-filter>
					<action android:name="android.intent.action.MAIN" />
					<category android:name="android.intent.category.LAUNCHER" />
				</intent-filter>
			</activity>
			<activity android:name=".indoorpositioning"
				android:label="Smart Uji Indoor Positioning">
				<intent-filter>
					<action android:name="android.intent.action.indoorpositioning" />
					<category android:name="android.intent.category.DEFAULT" />
				</intent-filter>
			</activity>
		</application>

	<uses-permission android:name="android.permission.INTERNET" />
	<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
	<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
</manifest>
```